Assistive System for Visually Impaired using Object Recognition

A thesis submitted in partial fulfilment of the requirement for the degree of

Master of Technology

in

Signal and Image Processing

by

Rahul Kumar
213EC6264

under the guidance of

Prof.(Dr.) Sukadev Meher

Department of Electronics and Communication Engineering

National Institute of Technology Rourkela

Rourkela, Odisha-769 008, India

May 2015

Assistive System for Visually Impaired using Object Recognition

A thesis submitted in partial fulfilment of the requirement for the degree of

Master of Technology

in

Signal and Image Processing

by

Rahul Kumar
213EC6264

under the guidance of

Prof.(Dr.) Sukadev Meher

Department of Electronics and Communication Engineering

National Institute of Technology Rourkela

Rourkela, Odisha-769 008, India

May 2015

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Certificate

This is to certify that the thesis titled, **"Assistive System for Visually Impaired using Object Recognition"**, submitted by **Mr. Rahul Kumar** bearing **Roll No. 213EC6264** in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Electronics and Communication Engineering** with specialization in **"Signal and Image Processing"** during session 2014-2015 at National Institute of Technology Rourkela is an original research work carried out under my supervision and guidance.

Prof. Sukadev Meher

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Declaration

I certify that,

- The work presented in this thesis is an original content of the research done by myself under the general supervision of my supervisor.
- The project work or any part of it has not been submitted to any other institute for any degree or diploma.
- I have followed the guidelines prescribed by the Institute in writing my thesis.
- I have given due credit to the materials (data, theoretical analysis and text) used by me from other sources by citing them wherever I used them and given their details in the references.
- I have given due credit to the sources (written material) used by quoting them where I used them and have cited those sources. Also their details are mentioned in the references.

Rahul Kumar

Acknowledgments

With immense pleaser I would like to express my deep gratitude to my supervisor, Prof. Sukadev Meher for his support and guidance throughout the work and providing lab facility and equipments to complete this work. His profound insights, exalting knowledge and valuable suggestions truly inspire and help me to complete this research work.

I am immensely indebted to Prof. Manish Okade for his comments and encouragement at different stages of the thesis, which were indeed thought provoking. My special thanks go to Prof. Ajit Kumar Sahoo, Prof. Lakshi Prosad Roy, Prof. Samit Ari for contributing towards enhancing the quality of the work, in shaping this thesis and teaching me subjects that proved to be very helpful in my work. I owe sincere thanks to Prof. Kamala Kanta Mohapatra for providing us financial support and foster the research environment throughout the work.

I would like to thank Deepak Kumar Panda, Sonia Das and Deepak Singh for there support and coordination throughout these years. I want to thank all my friends and lab–mates for their encouragement and cooperation. Their help can never be penned with words. Most importantly, none of this would have been possible without the love and patience of my family. This dissertation is dedicated to my beloved parents , they have been a constant source of love, concern, support and strength all these years. I am very grateful to the almighty God who is source of energy within me.

Abstract

Object recognition based electronic aid is most promising for visually impaired people to get description of nearby objects. With the advancement in computer vision and computing technologies we can afford to develop a system for visually impaired people, which can give audio feedback of surrounding objects and context. This thesis explore object recognition method to assist visually impaired people. We proposed an object recognition algorithm, and an assistive system which is very useful for their safety, quality life and freedom from other person all the time. Consideration of Gabor-recursive neural network along with convolutional recursive neural network(CRNN) with less number of maps have been found to be very promising to achieve better accuracy with less time complexity as compare to CRNN, the extracted feature vector is used to train Softmax classifier which is then used to classify query image into one of the trained categories. Our color recognition algorithm is simple and fast, which is the desiderata of color recognition module, we are using HSI color space with observed threshold and random sampling. The second contribution of this thesis is to use above stated methods to assist visually impaired people, the proposed assistive system is ensemble of two modules (i) object and (ii) color recognition, it is implemented using multimedia processor equipped embedded board and OpenCV. Object recognition module recognize objects such as door, chair, stairs, mobile phone etc. and generate an audio feedback to the user. Color recognition module generates audio description about object color in front of the camera, it is useful in recognizing clothing colors, fruits color etc. The system modules and operation can be selected using an on demand push button panel which contains two push buttons. The object recognition algorithm is evaluated on on-line available dataset as well as on our dataset and compared with state of art methods.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The environment around us is highly complex so we need several sensors such as vision, touch, smell etc. to survive in this world. All the creatures on Earth have a set of such sensors which help them searching food, water, and safety etc. Among all these sensors vision is critically important sensor because it give accurate and complex representation of the environment which can be processed to get valuable information. The main advantages of vision sensor as compare to other are its large and wide range, ability to provide complex data which can be processed to extract information such as object color, shape etc. In figure 1.1 some creatures are depicted, note that all of these are having a pair of eyes irrespective of environment(air, land, water), it can be concluded from the figure that, vision is primary sensor to survive not only on land but in water and air also. Human beings are highly dependent on vision sensor for daily tasks such as walking, eating, finding food, searching, driving vehicle, reading book etc., object recognition is the core algorithm in most of vision related task. Human outperform best computer vision algorithm to almost any measure and due to this main stream computer vision is always inspired from human vision, but visual neuroscience is limited to early vision [3] . We do object recognition all the time for example while you are reading this thesis you are recognizing characters and hence words, object recognition is needed in navigation, tracking, automation and so on, from above discussion we can convince ourself that object recognition is highly important. But what if someone born without vision capability or in some accident one lose sight?. Without eyes it

Figure 1.1: Some of the creatures on Earth with eyes.

is hard to survive even in his/her own environment, the person without sight has to memories everything in his environment and get irritated if someone misplace objects. Navigation and or searching objects in unfamiliar environment is surprisingly difficult for visually impaired people. In following sections we introduced object recognition, assistive technology and we cover work which have been done so far and related to our research. Motivation and problem statement is also given in this chapter.

1.1 Object Recognition

Object recognition is inherent part of our vision system to survive in this world, human and other creatures can perform this task instantly and effortlessly, but this is a hard problem for machine because each object in 3D world can cast infinite number of 2D

projections due to affine transforms, illumination change and camera viewpoint [4].

Object recognition is an extensively studied field, there are millions of articles available on object recognition, these articles can be divided into two broad categories one is shallow learning object recognition methods and other is deep learning object recognition methods. In the first approach hand designed features like SIFT [5], SURF [6], HOG [7] etc. are extracted in first stage, and typically next stages are bag of word and pyramid matching [8], the last stage of this pipeline is a classifier such as Support Vector Machine(SVM), K Nearest Neighbor(KNN), Artificial Neural network(ANN) etc. Though these methods are very effective for some applications but for generic object recognition hand design descriptor are not so effective due to high variation in shape, texture etc. The more recent approach to solve object recognition problem is using Deep learning based methods where a deep neural network is designed and then train it with millions of samples in a supervised manner [9], though it takes days or even weeks to train but it outperform all hand designed methods developed so far. Deep neural network architecture can be trained in two ways, one is using supervised learning method with millions of samples and second is using unsupervised learning method [10]. In unsupervised learning the network is trained with unlabeled data and it takes very less training time and labeled data is not required.

In our object recognition method, we are using convolutional neural network(CNN)[11] and recursive neural network(RNN)[12] in one pipeline, and Gabor feature extraction[13, 14] followed by pooling stage which goes to recursive neural network in second pipeline, the combined features vector is used to train Softmax classifier. Our method is capable of giving better accuracy with less number of features which is required for fast operation.

1.2 Assistive Technologies

Several assistive methods(vision substitution) have been developed so far, these methods can be categories as: rfid based methods, sonar based method, image processing based method and computer vision based methods. Among them computer vision

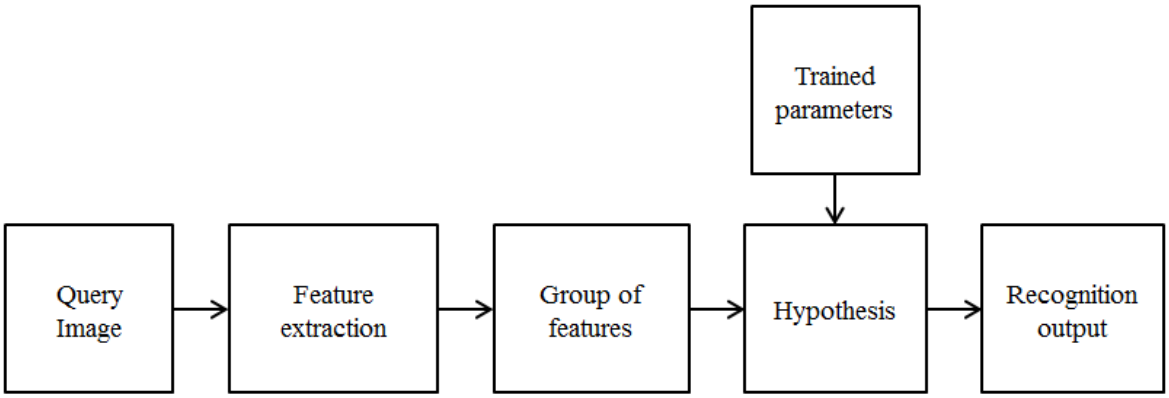


Figure 1.2: Typical object recognition pipeline.

based methods are most promising for object recognition application.

Visual impairment can be categorized into two parts [15] as below:

- Low vision is the case when visual acuity is less than 6/18 but more than or equal to 6/60.
- A person is considered as blind when visual acuity is less than 6/120.

We are all witness of technological advancement in recent years, which motivates us to develop a system for less privileged group of people. Several electronic assistive system has been developed so far, but there are very few which are using computer vision, but vision based systems are gaining momentum in recent research [16]. In figure 1.3 already developed assistive systems using computer vision is portrayed, figure 1.3(a) is a camera device which is developed to wear on finger and can be pointed in desired direction [17], in image figure 1.3(b) and (d) a camera mounted on glasses system is designed which works like an eye in between two eyes, so a person can rotate his head in potential object direction[18, 19]. In figure 1.3(c) and (f) a stereo vision cane is shown which employ 3D imaging to get depth data [20, 21]. Figure 1.3(e) depicted a virtual smart cane [22] which contains a laser and a smart phone with vibration mechanism. Some of these systems are developed for navigation and some are for particular application like bank note recognition as in 1.3(d).

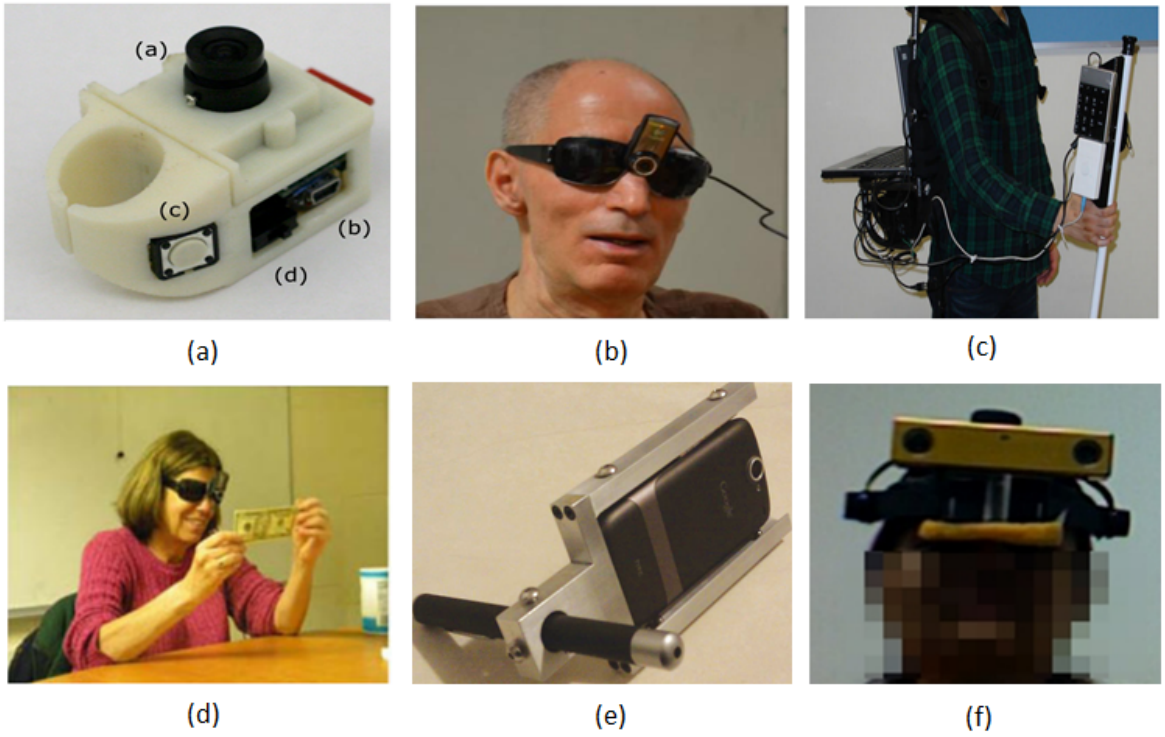


Figure 1.3: Assistive systems using computer vision.

1.3 Related Work

Researchers have developed assistive methods and devices to help visually impaired to provide safety and quality life. In this section the methods which incorporate computer vision technologies to assist visually impaired people are summarized, object recognition in particular. In Digital Object Recognition Audio (DORA) assistant for the visually impaired [23], the features such as brightness, color, and edge patterns are extracted from the captured image, the edge pattern is classified by artificial neural networks, though the parts of the system pertaining to brightness, color, and edge detection have been implemented, however, the object recognition system is under development. In [24] a comprehensive assistive system is proposed, in this method a smart phone with 3G network or wireless connection is used for live streaming with the host computer which process the data for object recognition and then the processed results are returned to mobile device, which goes to text to speech engine to generate audio feedback to blind user. This method work in real time but user working area is constrained by wireless signal strength. Another system is toward real-time grocery detection for the visually impaired [25], in this contribution a SURF descriptors is ex-

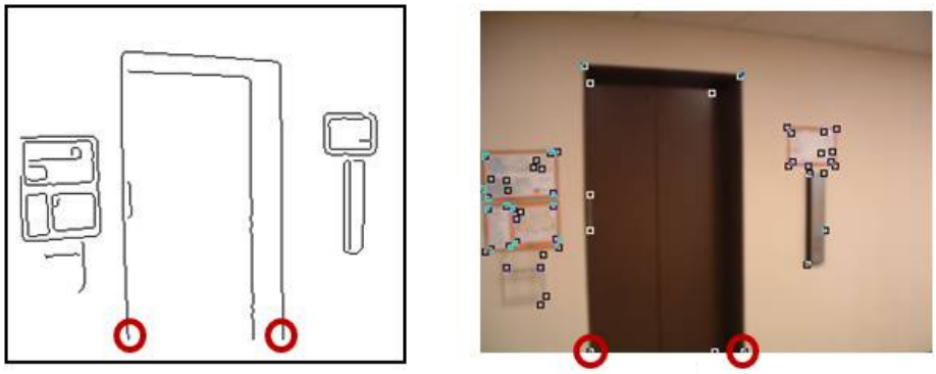


Figure 1.4: Edge and corner detected[1].

tracted from the image and a color histogram is also calculated at the location of each descriptor which is appended at the end of the descriptor, for object classification a naive-Bayes classifier is used. Though this method works with large number of categories but it does not operate in real-time. In [1] an algorithm is developed for door detection which is developed to work in an unfamiliar environment, in this method corner and edge features are used. In this method canny edge detector is used to get edge map of the preprocessed grayscale image, and then the corner detection method based on global and local curvature properties is used to detect corners as shown in 1.4, this figure is taken from the paper. In [26] a low cost system is developed which is using RFID reader and quick response (QR) code reader, this system is developed so that a visually impaired person could do shopping without any other person. In this system the RFID reader is placed on tip of a white cane so it can guide blind user in right direction and QR code(attached to shelf) is recognized by an object recognition algorithm. The method is useful only in pre-modified environment. In [20] a 3D scheme is used to recognize chair and stair objects, in this work a 3D kinect sensor from Microsoft is used to generate 3D data, entire system is shown in figure 1.3(c), this method generate vibration feedback when the desired object is found, the image processing is done on a laptop which makes the overall system bulky. A bank note recognition scheme is developed in [19], in this work a component based framework is proposed which uses SURF descriptor to describe each component of bank note, the system operational steps are shown in figure 1.5. The training step of this method is done on bank note ground truth, features are extracted and store in database for each category and its components, the components are defined based on its discriminative

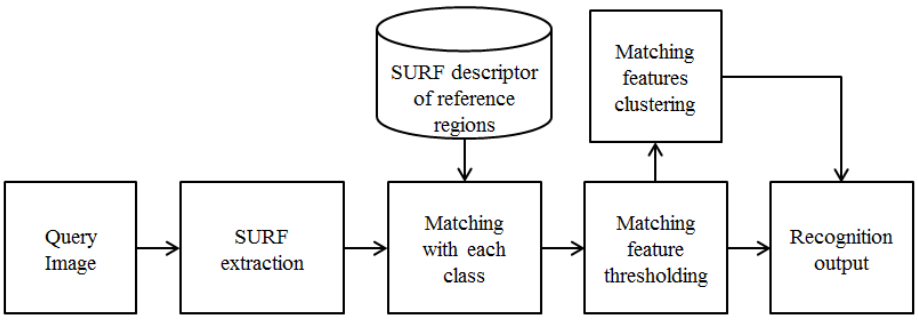


Figure 1.5: Bank note recognition system block diagram.

capability. For query image SURF features extracted and matched with the database, which goes to automatic thresholding and then matching to reduce falls negative is done. In [27] a method is developed to find lost objects, in this work Scale Invariant Feature Transform(SIFT) descriptor is used in combination with color attributes with sonification, sonification is used to guide a person's hand towards the query object. SIFT is used to locate predefined object patterns but for unknown object patterns color attribute is used for searching. Stairs and pedestrian detected and recognized in [28] using RGBD 3D camera is developed, the first stage is Hough transform to extract parallel lines from RGB image, depth is used for recognition, the linear SVM classifier is used to classify positive sample into one of the class i.e. stair or pedestrian and produces audio feedback to the blind user. In this method up and down stairs is recognized from the fact that up stairs have increasing steps and down stairs have decreasing steps in depth map but pedestrian have smooth depth change. In [2] restroom signage detection and recognition system is developed, in this method, first the attended area is extracted which may contain signage, it is being done by shape features, and then SIFT features are extracted to recognize the signage by passing the matching score through thresholding block. The recognition results are shown in figure 1.6.

In [29] a bus detection and recognition system is proposed, the system is able to recognize coming bus route and other text and generate speech to the visually impaired user, the bus detection is done by using HOG descriptor and cascade SVM classifier as shown in figure 1.7, and then text detection and recognition algorithm come into picture, its output goes to text to speech generation engine. A clothing pattern recognition prototype system is developed in [18], this system is able to recognize four clothing

Figure 1.6: Signage recognition, left to right: men, men with disability, women and women with disability [2].

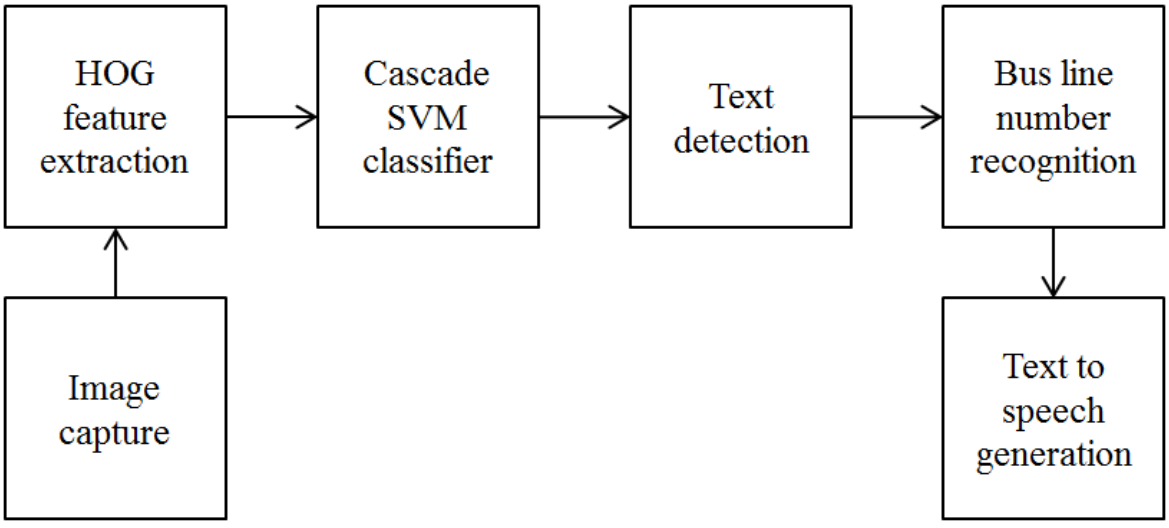


Figure 1.7: Bus detection system.

patterns and eleven clothing colors, the system produces audio feedback to visually impaired user through a Bluetooth earpiece and one more feature is there in this system that is it can be controlled by speech input, in this work random signature descriptor is proposed to extract clothing patterns. In [30] an integrated object recognition and location based services for the blind is developed, they are using template matching technique to recognition objects in the pathway. The technique is simple but time consuming and it fails when there is a big mismatch between the template's size and Pose.

1.4 Thesis Motivation

In a recent survey it is estimated that 285 million people worldwide are visually impaired, among them 246 million have low vision and 39 million are blind [31]. About 65% of all people who are visually impaired are aged 50 and older age group, with an increasing elderly population in many countries, more people will be at risk of visual impairment due to chronic eye diseases and ageing processes. More surprisingly, 19 million children are estimated as visually impaired and among them 1.4 million are irreversibly blind. It is estimated that approximately 90% of the visually impaired people are live in developing countries like China, India and so on. The facts about visual impairment is surprising on one hand, because despite of enormous medical efforts have been made to wipe out the visual impairment, the number is still breath taking. On the other hand these facts encourage us to develop object and color recognition based assistance for visually impaired people.

1.5 Thesis Objective

The object recognition is a challenging problem and needed to perform almost every task in this world, hence in this work we emphasis on object recognition based assistive method. Apart from object recognition, an explicit color recognition module can provide quick color information about the object in front of the camera.

The objective of the thesis is:

- To develop object recognition algorithm.
- To develop a system using object and color recognition which could enhance the capability of visually impaired people without overriding their auditory capability.

1.6 Thesis Organization

The thesis is consists of five chapters as described below:

Chapter 1: we introduce the object recognition and its challenges in this chapter, apart from this, we have discussed several assistive methods using computer vision being used to assist visually impaired people and represent their pros and cons.

Chapter 2: Feature extraction is crucial block in object recognition pipeline as shown in figure 1.2. In this chapter we elaborate feature extraction and learning, for object recognition and we also discus color descriptor to extract object color attributes. Pre-processing and local contrast normalization is also discussed.

Chapter 3: In this chapter object recognition algorithm is proposed. To evaluate the algorithm a household objects database is made, the dataset contains images of objects such as door, mobile phone, chair etc. with changes such as view points, scaling and illumination etc. Object recognition algorithm results are compared with other methods and analysis results are also depicted.

Chapter 4: The system is implemented using Raspberry Pi Hardware, the implementation details are discussed. Object and color recognition results with processing time are portrayed.

Chapter 5: Thesis is concluded and future scope is discussed in this chapter.

Chapter 2

Feature Extraction and Feature Learning

There are two ways to describe an object one is by using hand designed feature descriptor and the second is to learn features from the dataset itself, for specific applications with deterministic conditions hand designed descriptors works very well, for uncertain and changing environment its is very hard to predict every possible case(like in object recognition) and hence feature learning works better. In the following sections hand designed and self learning feature descriptors are described.

2.1 Hand designed Descriptors

If we take n number of images of same object than we can observe that the images are not same, it may be due to illumination change, scaling, camera jitter, rotation, camera noise etc., another problem with the object image is that, it's very rare to find single object in an image, there may be several other unwanted objects along with the desired object and hence we can not use direct captured images by camera to classify the object. Every object have some feature such as set of corners, edges, color etc. In following subsections we will discus few very popular descriptors.

2.1.1 SIFT

Scale invariant feature transform(SIFT) [5] is very popular and extensively used descriptor. SIFT is invariant to image scale and rotation which make it suitable for real world applications. It is a local feature extractor so robust to clutter and occlusion, this property is very desirable for a descriptor, since it is of practical demand to identify object under conditions such as occlusion and clutter. The other advantage of SIFT descriptor is that it is close to real time performance, so one can use it for commercial applications. SIFT is a four step algorithm as shown below:

- Scale space peak selection.
- Key point localization.
- Orientation assignment.
- Key point descriptor.

To describe an object SIFT first identify the stable keypoints which are invariant and then apply descriptor to describe the object on those point. The first step is to find out interest points, which are nothing but local maxima in scale space of Laplacian of Gaussian(LOG). LOG is applied to an image with different sigma values resulting several images of different-different scales, now to identify the interest point a pixel of 3x3 window is taken and then check it with its neighborhood on that scale, scale above and below it so this pixel is being compare with 26 neighboring pixels for extrema, if the pixel is extrema(minima/maxima) then that point is potential interest point.

LOG can be approximated by Difference of Gaussian(DOG) as in equation 2.1 which is used by SIFT to evaluate LOG. DOG is nothing but apply a Gaussian filter with σ value and then apply another Gaussian filter with $k\sigma$ value and then subtract one from another, the result is DOG as in equation 2.2 and 2.3, DOG is evaluated for each octave. From the empirical study done by D.Love[5] 3 scale in each octave is pretty good choice and sigma as 1.6 for high repeatability.

$$\frac{\partial G}{\partial \sigma} = \sigma \Delta^2 G \tag{2.1}$$

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2\Delta^2 G \tag{2.2}$$

$$G(x, y, k\sigma) = \frac{1}{2\pi(k\sigma)^2} e^{-(x^2+y^2)/2k^2\sigma^2} \tag{2.3}$$

with the procedure described above large number of extrema will be generated, among them most stable extrema detected and this process is called key point localization.

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X \tag{2.4}$$

where $X = (x, y, \sigma)^T$.

Extrema is located at

$$\hat{X} = -\frac{\partial^2 D^{-}1}{\partial X^2} \frac{\partial D}{\partial X} \tag{2.5}$$

The value of D(x) at extrema must be larger than a threshold value i.e. mod $D(x) = th$. After initial outliers reject step further outliers rejection is done. DOG has strong response along edge. Compute Hessian of D.

$$H = \begin{bmatrix} D_1 & D_2 \\ D_3 & D_4 \end{bmatrix} \tag{2.6}$$

Now remove outliers by using equation 2.7

$$\frac{T_r(H)^2}{Det(H)} = \frac{(r+1)^2}{r} \tag{2.7}$$

Where $r = D1/D4$ and $Det(H) = D1 \times D2$. The key-points having r value greater than 10 are rejected.

The next step is orientation assignment to achieve rotation invariance. Compute central derivative, gradient and direction of smooth image at scale of interest point(x,y).

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \tag{2.8}$$

$$\theta(x, y) = \arctan\left(L(x, y+1) - L(x, y-1)\right)/(L(x+1, y) - L(x-1, y)) \tag{2.9}$$

2.1.2 Fourier Descriptor

Fourier descriptor is another hand designed feature extractor, it is used to describe object shape, it is a boundary based descriptor, it uses Fourier coefficients to approximate the object shape. The object boundary can be expressed as in equation 2.10.

$$s(k) = x(k) + jy(k) \tag{2.10}$$

where $k = 0, 1, 2, 3, ..., K - 1$ and K are the number of points on boundary. Now we can take DFT of s(k) as below

$$S(u) = \sum_{k=0}^{K-1} s(k) e^{-j2\pi uk/K} \tag{2.11}$$

where $u = 0, 1, 2, 3, ..., K - 1$. The complex coefficient S(u) are nothing but Fourier descriptors. Now to retain the boundary from these Fourier descriptors inverse Fourier transform is used as below

$$s(k) = \frac{1}{K} \sum_{u=0}^{K-1} S(u) e^{j2\pi uk/K} \tag{2.12}$$

But we can approximate the object structure using less then K Fourier coefficient, for example $u = 0, 1, 2, ..., P$ where P is less than K.

2.1.3 Color Descriptors

Every object in the world carry some color with it, so color is important attribute to know about any object. But object color is highly dependent on environmental conditions such as illumination etc. So we need some color descriptor which can reliably describe the object color irrespective of illumination change. The change in illumination can be modeled by the diagonal offset model proposed in [32].

$$\begin{bmatrix} R^c \\ G^c \\ B^c \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} R^u \\ G^u \\ B^u \end{bmatrix} + \begin{bmatrix} o_1 \\ o_2 \\ o_3 \end{bmatrix} \tag{2.13}$$

In diagonal offset model super script u stands for unknown light and c is for canonical illumination. The second term in right hand side is offset and diagonal matrix containing a, b, c is used to map colors which are taken in unknown condition to canonical. On the basis of a, b, c values and offset we can classify the change occurred in the image into five ways . if $a = b = c$ and no offset then its is only light intensity change, If $a = b = c = 1$ and $o_1 = o_2 = o_3$ this is called light intensity shift which results equal shift in intensity in all channels. Diffuse light results intensity sift. Third case is where both light intensity change and shift comes into picture i.e. $a = b = c$ and $o_1 = o_2 = o_3$. The fourth case is full diagonal model where $a \neq b \neq c$ and no offset, this class of change can model light scattering and illumination color change. Fifth case is full diagonal and offset where $a \neq b \neq c$ and $o_1 \neq o_2 \neq o_3$, this change is called light color change and shift.

From the above discussion we can say that in order to describe the object color we need some kind of descriptor which does not affect or little affect by the changes discussed. we are describing some of the color descriptors below:

Opponent Histogram: It is a combination of three one dimensional histograms which is based on the opponent color space channels.

$$\begin{bmatrix} O_1 \\ O_2 \\ O_3 \end{bmatrix} = \begin{bmatrix} \frac{R-G}{\sqrt{2}} \\ \\ \frac{R+G-2B}{\sqrt{6}} \\ \\ \frac{R+G+B}{\sqrt{3}} \end{bmatrix} \tag{2.14}$$

O_1 and O_2 represent color information and O_3 represent intensity of the image, one can observe the subtraction in O_1 and O_2 which results shift invariance with respect to light intensity but O_3 has no such property.

Hue Histogram: Hue color model is invariant to scale and shift with respect to light intensity change. Hue instability could be counter by weighting hue by saturation.

$$hue = \arctan \frac{\sqrt{3}(R-G)}{R+G-2B} \tag{2.15}$$

Opponent derivative descriptor: The opponent angle is invariant to diffusion

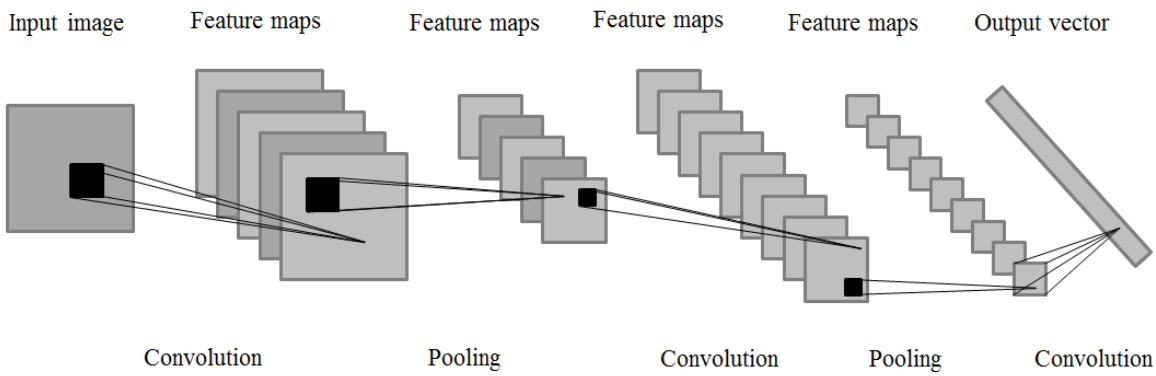


Figure 2.1: Convolutional neural network.

lighting.

$$angle = \arctan \frac{O_y}{O_x} \tag{2.16}$$

Where O_y and O_x are spatial derivatives in chromatic opponent channels.

2.2 Convolutional Neural Network

Convolutional neural network(CNN) is designed to recognize 2D shapes irrespective of object translation, rotation, scaling etc. CNN is neurobiologically motivated, it is first proposed by LeCun, the basic form of convolutional neural network consists of three main operations as follows:

- feature extraction
- Feature mapping
- Subsampling

The structure shown in figure 2.1 is having consecutive convolution and pooling stages, which is inspired from simple cell and complex cell in our brain. Feature extraction is done by neurons taking synaptic input from local receptive field from the previous layer and hence it extract local features from the image, one important point to note with CNN is that all neurons in one maps share same synaptic weights which drastically reduce number of free parameters. Once a feature is extracted its exact location become irrelevant as far as its relative position preserves.

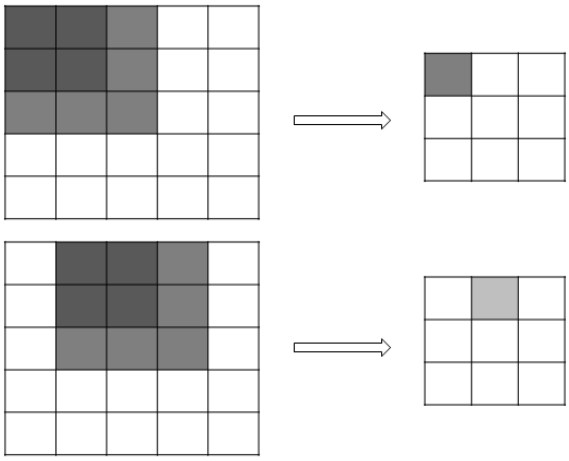


Figure 2.2: Convolution operation with some filter.

Another important operation is feature mapping, each computational layer of CNN contains multiple feature maps, where each map is of the form of a plane. Each neuron in a plane share same set of synaptic weights as said earlier, this structure enjoy advantages like shift invariance and reduction in the number of free parameters which need to be trained.

$$v_j^n = \sum_{k=1}^{C} w_{kj}^n * v_k^{n-1} \tag{2.17}$$

$$v_j^{n+1} = max(0, v_j^n) \tag{2.18}$$

Where $j = 0, 1, 2, 3, ...K$, C is 3 for color images. v_j^n is j^{th} output map and v_k^{n-1} is input k^{th} map which is being convolved with filter w_{kj}^n, $*$ is convolution operation.

If input image is a color image then each filter is a 3D or 3 channel, it means first channel of filter will convolved with red channel of the image, second channel of the filter is convolved with second channel of the image and so on, the convolved result of each channel then sum to produce one output map corresponding to that filter. Now second filter produces second map in the same way and so on. The convolution operation could be well understood by the figures 2.2 and 2.3. Figure 2.2 is depicting convolution operation to produce first map, from this its is clear that to get each pixel in output map same filter is being used. Figure 2.2 portray convolution of first 3×3 filter with a given 5×5 image to produce first output map, in figure 2.3 second filter of same size is convolving with same image to produce second output map. So to produce K number of maps in any hidden layer of CNN, K number of filters are required.

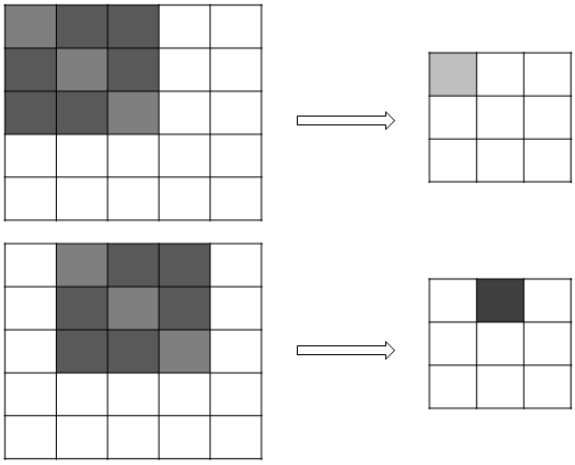


Figure 2.3: Convolution operation with some filter.

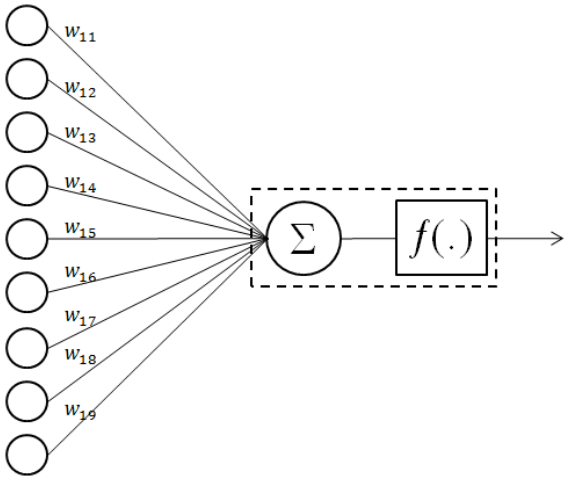


Figure 2.4: One neuron of CNN.

The convolution operation is valid convolution and hence the output map size would be $(d_i^1 - d_f^1 + 1) \times (d_i^2 - d_f^2 + 1)$. Where $d_i^1 \times d_i^2$ is input image size and $d_f^1 \times d_f^2$ is filter size, note that we are explaining the concept for 2D case which can be easily extended to 3D case.

As we get convolved output it need to passed through nonlinearity such as *sigmoid*, *tanh*, *ReLu* etc. Figure2.4 is one neuron which take 3×3 receptive field and produce some excitation depending upon trained weights. In the figure2.4, f generally any nonlinearity function like *tanh* etc and w_{11}, w_{12}, w_{13} etc. are trained weights which are being multiplied with input image pixel x_1, x_2, x_3 etc.

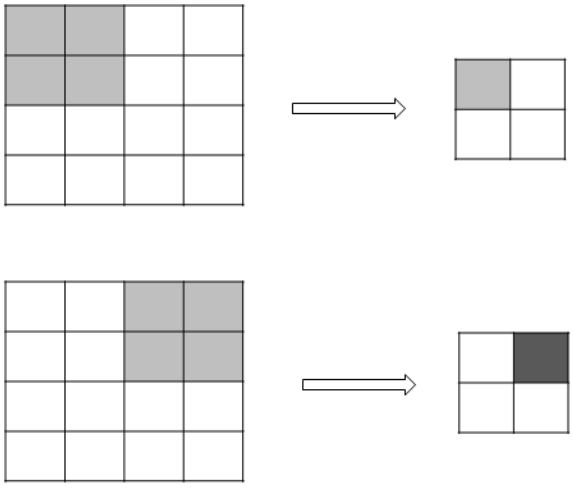


Figure 2.5: Pooling with 2×2 window size.

After performing convolution operation next step is pooling of maps, there are many type of pooling operations such as average pooling, max pooling, L2 pooling etc. pooling reduce the dimension of maps and it reduce the sensitivity to shift and other form of distortions.

$$v_j^{n+3}(p,q) = \frac{1}{N}\left(\sum_{\bar{p}\in\mathcal{N}(p),\bar{q}\in\mathcal{N}(q)} v_j^{n+2}(\bar{p},\bar{q})\right) \tag{2.19}$$

$$v_j^{n+3}(p,q) = max_{\bar{p}\in\mathcal{N}(p),\bar{q}\in\mathcal{N}(q)} v_j^{n+2}(\bar{p},\bar{q}) \tag{2.20}$$

$$v_j^{n+3}(p,q) = \sqrt{\sum_{\bar{p}\in\mathcal{N}(p),\bar{q}\in\mathcal{N}(q)} (v_j^{n+2}(\bar{p},\bar{q}))^2} \tag{2.21}$$

Where $\mathcal{N}$ is number of neighborhood for pixel at position (p,q) which is used for pooling. Equation 2.19, 2.20 and 2.21 are average pooling, max pooling and L2 polling respectively.

In pooling the window size may vary it depends on application, generally the stride in pooling stage is equal to window size but it is not necessary, the output image of pooling stage, is of the size $((d_m^1 - d_w^1)/s + 1) \times ((d_m^2 - d_w^2)/s + 1)$. where $d_m^1 \times d_m^2$ is map size, $d_w^1 \times d_w^2$ is window size and s is stride.

Apart from convolution and pooling stage researchers have observed that Local contrast normalization(LCN) is also very useful for applications like object recognition. LCN is biologically inspired, it provide invariance of the features with respect to intensity change. In figure 2.6 image processed by LCN operation is shown, figure 2.6(a) is original image with very poor intensity condition, in figure 2.6(b) LCN processed image is shown, in which the object is much more illuminated then the original image, in figure 2.6(c)(d)(e) red, green and blue channels respectively of original image are portrayed, Local Contrast Normalization processed R,G,B channels are depicted in 2.6(f), (g) and (h).

$$v_j^{n+2}(p,q) = v_j^{n+1}(p,q) - \sum_{k,\bar{p}\in\mathcal{N}(p),\bar{q}\in\mathcal{N}(q)} w_{\bar{p}\bar{q}}.v_k^{n+2}(\bar{p},\bar{q}) \tag{2.22}$$

where k=0,1,2,...K.

$$v_j^{n+3}(p,q) = \frac{v_j^{n+2}(p,q)}{max(\epsilon, \sigma^{n+1}(\mathcal{N}(p,q)))} \tag{2.23}$$

where

$$\sigma^{n+1}(\mathcal{N}(p,q)) = \sqrt{\sum_{k,\bar{p}\in\mathcal{N}(p),\bar{q}\in\mathcal{N}(q)} w_{\bar{p}\bar{q}}.(v_k^{n+2}(\bar{p},\bar{q}))^2} \tag{2.24}$$

Now the question is how to train this network? , there are two ways one is supervised learning method where a large number of labeled samples are given to train the network using Back-propagation algorithm. The second method is unsupervised learning method, this method is recently introduced and shown good results with few examples used for training. The weights of network can be trained in unsupervised manner from the unlabeled dataset. Researchers have introduced several algorithms to train network in unsupervised manner such as Auto-encoder, GMM, K-means clustering etc. In section2.5 and 2.6 we elaborate two commonly used unsupervised learning algorithm to train Convolutional neural networks.

In figure 2.7 original image and processed image at different stages of convolutional neural network is depicted. Figure 2.7(a) is original image which is convolved with one of the trained filter(3^{rd} filter in top row in figure2.12), the output is shown in figure 2.7(b), its is pass through non linearity to get figure 2.7(c), output of nonlinearity is

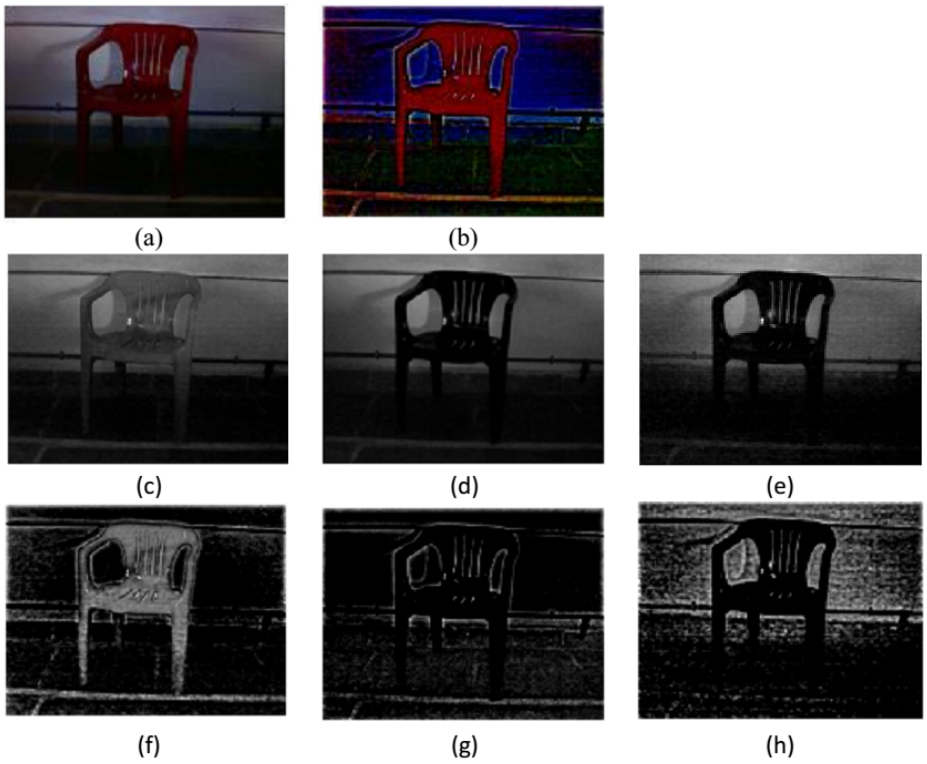


Figure 2.6: Local contrast normalization.

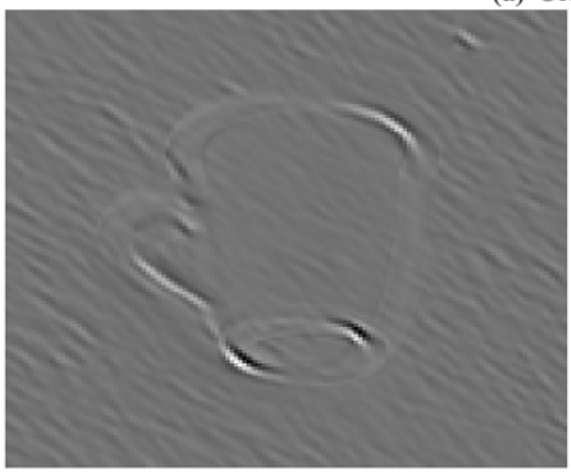
processed for Local contrast normalization and Pooling, output of LCN and Pooling is depicted in figure 2.7(d) and (e).

2.3 Recursive neural network

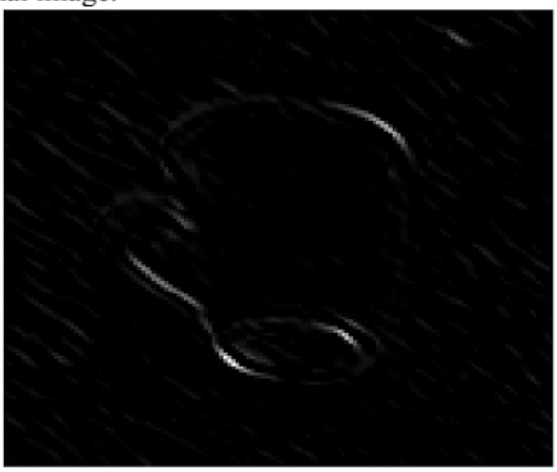
Recursive neural network is used in this work to extract hierarchical features which is the recursion of same network, the leaf node are nothing but K dimensional feature vectors from the output of CNN or from the output of pooling stage. We use fixed tree multiple RNN to extract features as explained in [12]. The neural network to compute parent node vector is as below:

$$p = f\left(W \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_{b^2} \end{bmatrix}\right) \tag{2.25}$$

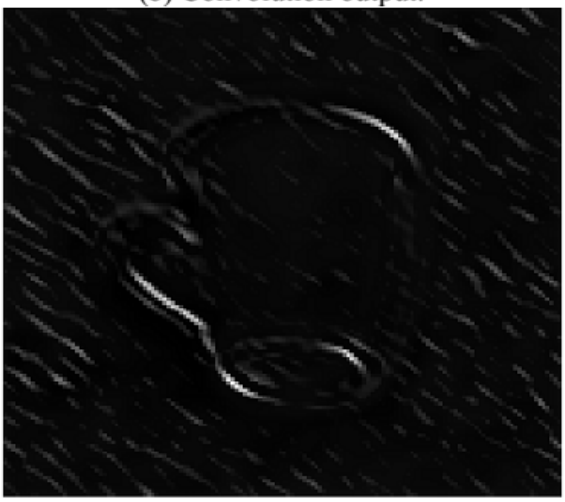
(a) Original image.



(b) Convolution output.



(c) Output of ReLU.



(d) Output of LCN.

(e) Output of Pooling stage.

Figure 2.7: Processed image of each stage in CNN.

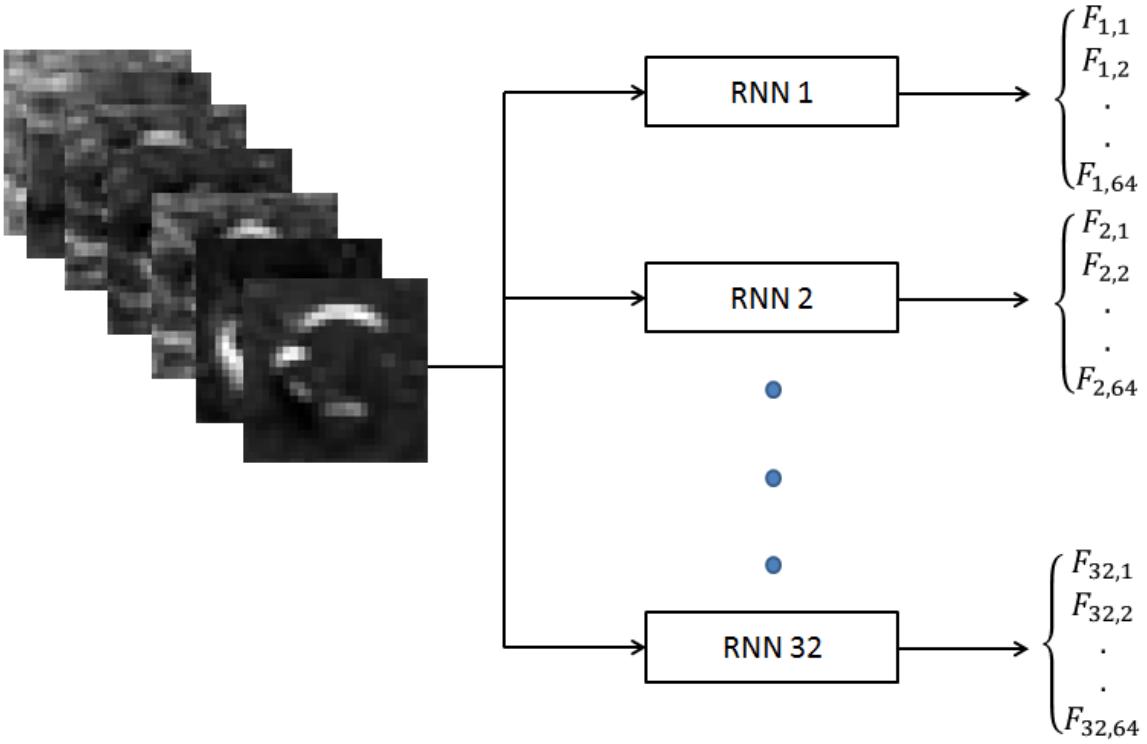


Figure 2.8: Multiple RNN on output maps of CNN.

Where W is the weight of neural network, and note that W is same for whole tree, f is non-linearity function such as $tanh$(2.26), $sigmoid$(2.27) etc, we are using $tanh$ in our experiment. One can can use any number of such RNN depending upon the application demand. As in figure 2.8, 32 RNN on 64 CNN maps is portrayed, where each RNN contains a fixed tree structure. This Multiple RNN arrangement would generate 2048 dimensional feature vector. In this structure every RNN have different weight eg. RNN1 is having different weight than RNN2, but inside one RNN the weights are exactly same.

$$f(y) = \frac{e^{y} - e^{-y}}{e^{y} + e^{-y}} \tag{2.26}$$

$$f(y) = \frac{1}{1 + e^{-y}} \tag{2.27}$$

2.4 Gabor Features

Gabor feature extractor is biologically inspired[3], we are extracting Gabor features from grayscale images, the Gabor filters are applied with S1 scales and O1 orientations, as in figure 2.9, 8 scale and 5 orientation gabor filters are depicted and filtered image of a coffee mug is depicted in figure 2.10.

$$G(x,y) = e^{-(x_o^2+\gamma^2 y_o^2)/2\sigma^2} xcos(\frac{2\pi}{\lambda}x_o) \tag{2.28}$$

Where $x_o = xcos(\theta) + ysin(\theta)$ and $y_o = -xsin(\theta) + ycos(\theta)$ and $\theta, \gamma, \sigma, \lambda$ are orientation,aspect ration,effective width and wavelength respectively.

Gabor filter is able to produce very accurate description of simple cells in ventral stream. Gabor function is nothing but multiplication of elliptical Gaussian pulse and sinusoid[13].

$$G_p(x,y) = e^{-(\frac{x_o^2}{a^2}+\frac{y_o^2}{b^2})/2} \tag{2.29}$$

Where a^2 and b^2 are variance in x and y direction respectively.

$x^{''} = x - x^{'}$ and $y^{''} = y - y^{'}$ are used to locate Gaussian pulse at location $x^{'}$ and $y^{'}$. The orientation can be controlled as follows:

$$x_0 = x^{''}cos(\theta) - y^{''}sin(\theta) \tag{2.30}$$

$$y_0 = -x^{''}sin(\theta) + y^{''}cos(\theta) \tag{2.31}$$

Now this elliptical Gaussian pulse is multiplied with sinusoids to get Gabor filter as in equation 2.28.

2.5 K-means clustering

K-means clustering is a well-known unsupervised clustering algorithm. The power of this algorithm resides in its simplicity, fast processing and convergence. The K-means

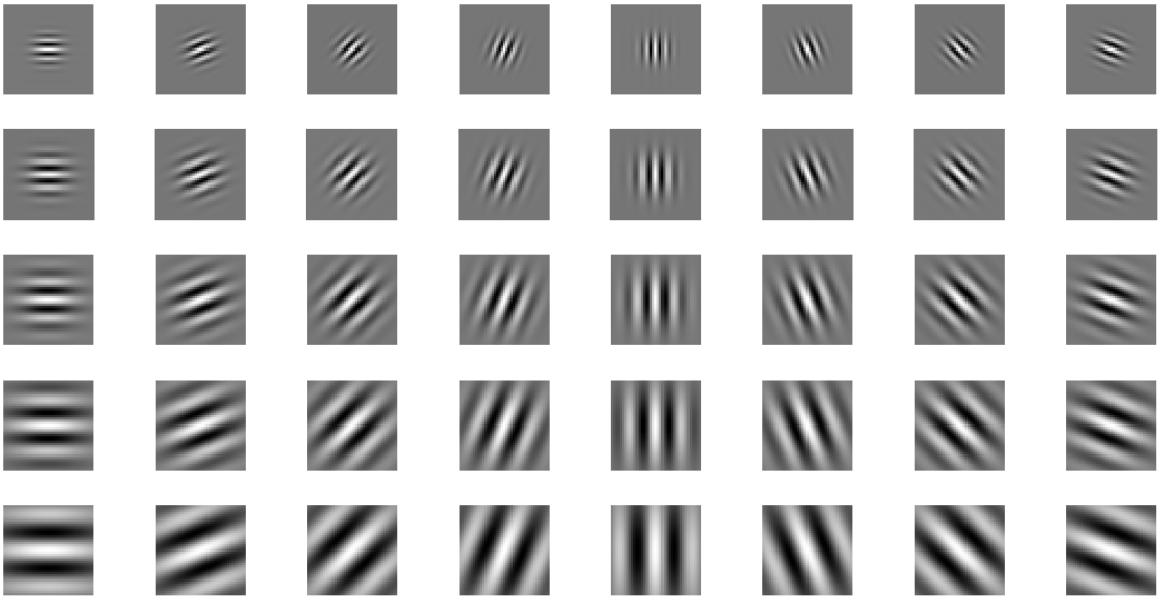


Figure 2.9: Gabor filters with 5 orientation and 8 scales.

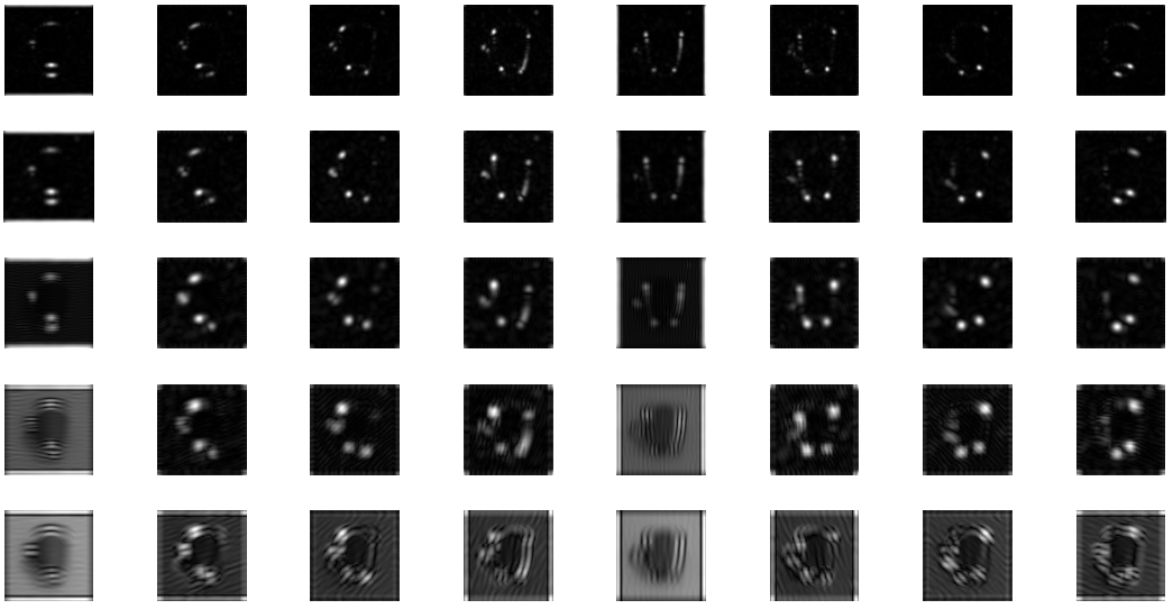


Figure 2.10: Gabor filter processed results on coffee mug.

clustering algorithm can be proceed as follows:

Step 1: Randomly initialized cluster centroids $\{\mu_j\}_{j=1}^{K}$.

Step 2: For every i,

$$c_i = \underset{j}{\operatorname{argmin}} \|x_i - \mu_j\|^2 \tag{2.32}$$

Step 3: For every j,

$$\mu_j = \frac{\sum_{i=1}^{m} t_{ij}.x_i}{\sum_{i=1}^{m} t_{ij}} \tag{2.33}$$

step 2 and step 3 are repeated until it converge.

$$t_{ij} = \begin{cases} t_{ij} = 1 & \text{if } c_i = j \\ t_{ij} = 0 & otherwise \end{cases}$$

The cost function for K-means algorithm is as below:

$$L = \sum_{j=1}^{K} \sum_{c_i=j} \|x_i - \mu_{c_j}\|^2 \tag{2.34}$$

where $j = 1, 2, 3, ..., K$ and $i = 1, 2, 3, ..., m$, K and m are are the number of clusters and samples respectively. The cost function is used to ensure the convergence of algorithm, as we can observe for above depicted steps that in first step fixed μ is used to assign the clusters under the constraint of minimization of squared euclidean distance of samples from the mean. In second step mean is computed for assigned clusters to minimize the cost function. This procedure repeat iteratively until cost function stop changing.

The above stated K-means algorithms can be extended to images as follows:

- Initialize centroids of required dimension.
- Extract the patches from unlabeled data.
- Pre-processing (Patches normalization and whitening).
- Label each patch with respect to its nearest centroid.
- Update centroids.

Repeat step 4^{th} and 5^{th} until algorithm converges. In this work we are using Kmeans clustering algorithm to train CNN in unsupervised manner, in figure 2.12 weights

trained using the algorithm above is depicted, from the weights we can observe that each neuron in the Convolutional neural network respond to specific pattern only.

2.5.1 Pre-processing

K-means clustering algorithm can not handle correlated data, and we know images have high degree of correlation so before applying K-means clustering algorithm, pre-processing needs to be done on the data, preprocessing steps are as below:

- Subtractive normalization.
- Divisive normalization.
- Whitening.

In figure 2.11 a two dimensional feature $X = \begin{bmatrix} x1 \\ x2 \end{bmatrix}$ is plotted, where $x1$ and $x2$ are having different variances and lying in first quadrant only. In figure 2.13 preprocessing is performed on image data, figure 2.13(a) is original image, 2.13(b), (c) and (d) are mean subtractive, variance normalized and whitened image respectively. Each operation is described in following sub sections.

Subtractive normalization

It is used to make data zero mean by subtraction of mean from the data, this step is very useful to make feature extractor robust to unknown data. After mean subtraction we can observe that features are in all four quadrant as in figure 2.11(b). In figure 2.13(b) subtractive preprocessing is portrayed on image data.

$$v_j^{n+2}(p,q) = v_j^{n+1}(p,q) - \sum_{k,\bar{p}\in\mathcal{N}(p),\bar{q}\in\mathcal{N}(q)} v_k^{n+1}(\bar{p},\bar{q}) \tag{2.35}$$

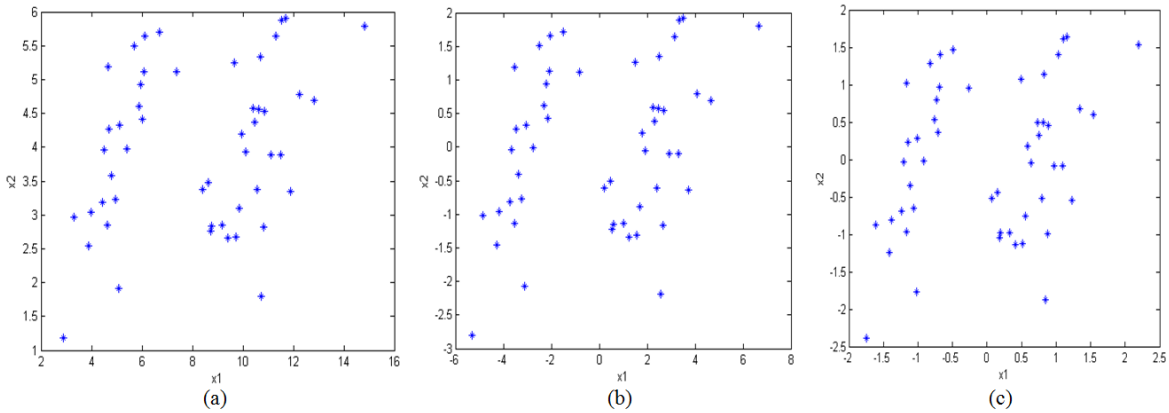


Figure 2.11: Data normalization.

Divisive normalization

This step makes uniform variance of all of the features. After standard deviation normalization we can observe from figure 2.11(c) that the variation of both of the features are in almost same range. In figure 2.13(c) variance normalization is portrayed on image data, the variance normalization is done on mean subtracted image.

$$v_j^{n+3}(p,q) = \frac{v_j^{n+2}(p,q)}{max(\epsilon, \sigma^{n+1}(\mathcal{N}(p,q)))} \tag{2.36}$$

where $\sigma^{n+1}(\mathcal{N}(p,q)) = \sqrt{\sum_{k,\bar{p}\in\mathcal{N}(p),\bar{q}\in\mathcal{N}(q)}(v_k^{n+2}(\bar{p},\bar{q}))^2}$ and ϵ is a small number.

Whitening

Whitening is the process to make data decorrelated. In this work we are using ZCA[33] transform for whitening. Lets W is decorrelation matrix. In figure 2.13(d) whitening preprocessing is portrayed on image data, which can be verified from the output. In figure 2.14 red, green and blue channels are depicted. In figure 2.14(a) and (b) original image and whitened images are shown respectively, figure2.14(c), (d), (e) are R, G, B channels of original image and figure 2.14(f), (g), (h) are whitened output of each channel are portrayed.

For zero phase whitening $W = W^T$ (symmetric) Hence $W = (XX^T)^{-1}$

$$Y = WX \tag{2.37}$$

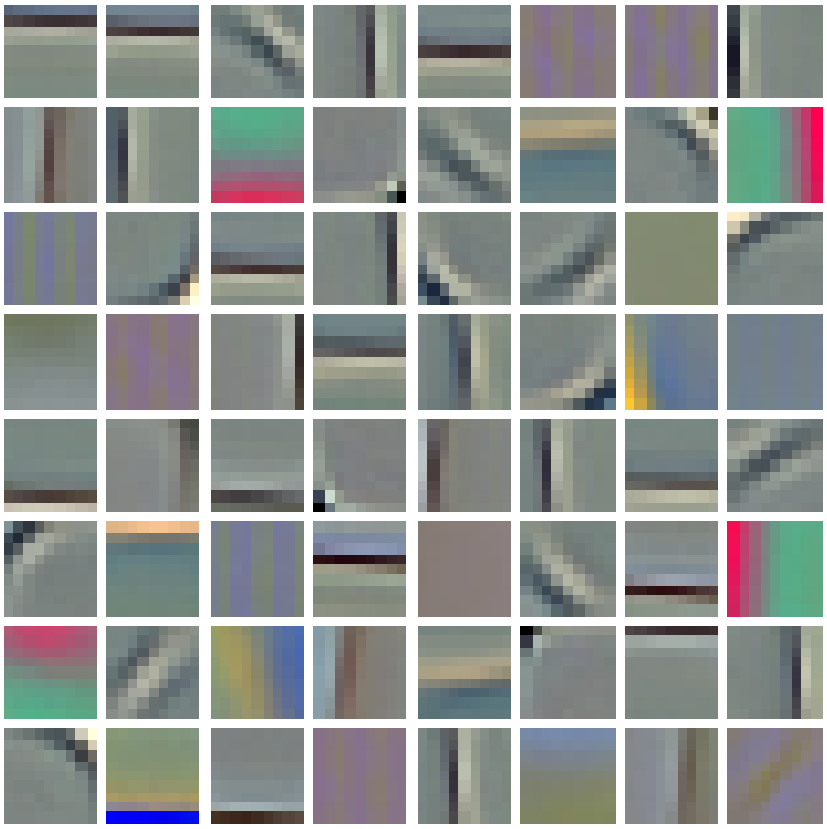


Figure 2.12: Trained filters using K-means algorithm.

$$W = VD^{-1/2}V^T \tag{2.38}$$

where V and D are calculated from the equation below:

$$[V, D] = eig(cov(X)) \tag{2.39}$$

where $cov(x_i, x_j) = E[(X_i - \mu_i)(X_j - \mu_j)]$. In equation 2.37 a small constant ϵ is also added for practical reasons, so the modified equation after adding new term is as in eq 2.40.

$$W = V[D + \epsilon I]^{-1/2}V^T \tag{2.40}$$

2.6 Autoencoder

Autoencoder is another method to train convolutional neural network in an unsupervised manner. Autoencoder is nothing but a multilayer perceptron(MLP)[34], the free

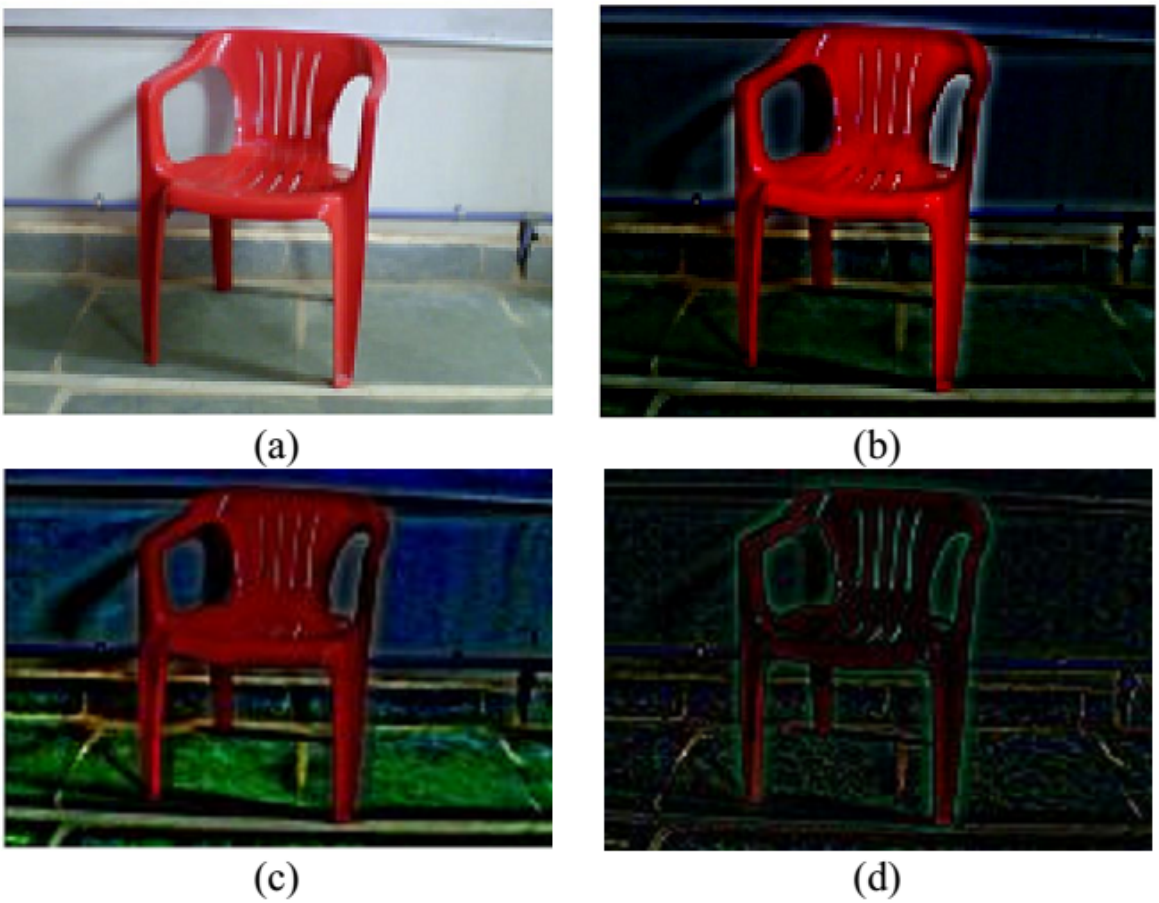


Figure 2.13: Image preprocessing (a) original image, (b) mean subtraction (c) Standard deviation normalization (d) whitening.

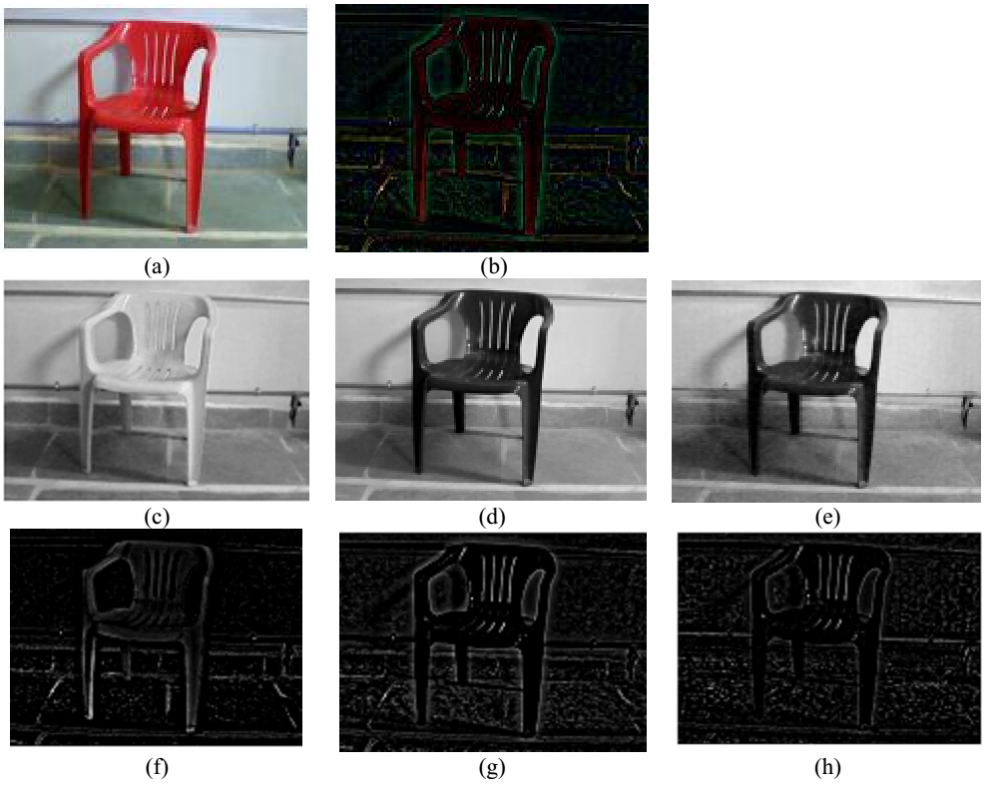


Figure 2.14: Image whitening using ZCA, midle row rgb channels and bottom row is whitened results of each channel.

parameters are being trained using back-propagation algorithm with random patches of size equal to receptive filed of convolutional neural network. The main difference between perceptron and autoencoder is in its target values, in MLP the target is nothing but object class labels but in case of Autoencoder the target is same as input layer i.e. $y_i = x_i$ as shown in figure 2.15. From the Figure 2.15 one can observe that number of units in layer L1 and Layer L3 are exactly equal, L2 is hidden layer. Autoencoder is basically try to match the input values by changing its weights with the help of backpropagation algorithm.

There are two processing steps in autoencoder one is forward propagation and second is back propagation. Forward propagation is done as in equations 2.41 and 2.42, in these equations we have depicted computation of one neuron in layer 1.

$$z_i^l = \sum_{j=1}^{n} w_{ij}^{l-1} x_j + b_i^{l-1} \tag{2.41}$$

$$a_i^l = f(z_i^l) \tag{2.42}$$

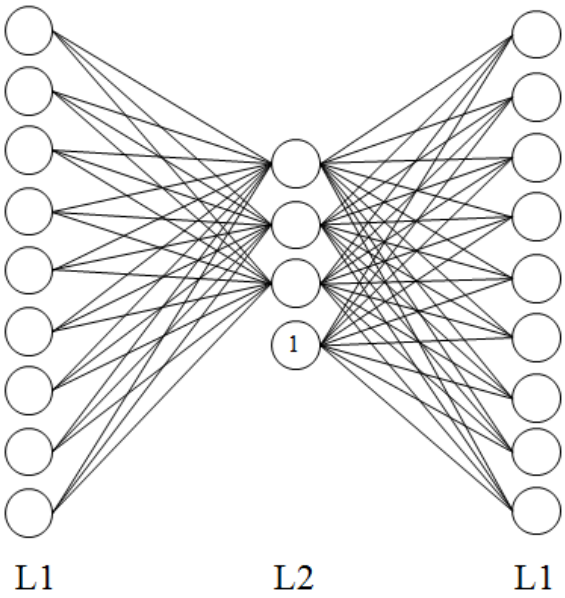


Figure 2.15: Autoencoder.

where f is nonlinearity function such as $sigmoid, tanh$ etc.

The second pass is back propagation of error, to train the weights which were randomly initialized with small values. The weights are then updated as in equation 2.43.

$$w_{ij}^{l} = w_{ij}^{l} - \alpha \frac{\partial J(w,b)}{\partial w_{ij}^{l}} \tag{2.43}$$

$$b_{i}^{l} = w_{i}^{l} - \alpha \frac{\partial J(w,b)}{\partial b_{i}^{l}} \tag{2.44}$$

now the partial derivatives can be effectively computed using backpropagation algorithm, for more details on back propagation refer to [35].

$$\frac{\partial J(w,b)}{\partial w_{ij}^{l}} = a_{i}^{l} \delta_{i}^{l+1} \tag{2.45}$$

$$\frac{\partial J(w,b)}{\partial b_{i}^{l}} = \delta_{i}^{l+1} \tag{2.46}$$

where δ_{i}^{l+1} is error at layer $l+1$ in i^{th} unit.

2.7 Summery

In this chapter we have explained hand design descriptors as well as feature learning method to extract features. In our object recognition module we are using biologically inspired methods such as CNN and Gabor. Apart from the conceptual explanation of CNN, LCN, normalization etc. we have portrayed output of these blocks processed in MATLAB software. Unsupervised learning is explained and filters trained with K-means clustering is depicted.

In next chapter we will use convolutional neural network with K-means clustering algorithm to train the CNN in unsupervised manner. We chose K-means clustering over Autoencoder due to its simplicity and processing speed.

Chapter 3

Object Recognition and Dataset

3.1 Dataset

In this section we discus dataset used to evaluate our algorithm, we have used two datasets, one is Household object dataset(Our) which contains images with several variations such as intensity, viewpoint, scale, rotation etc. and second is Mit indoor sub-dataset.

Our dataset We have develop a dataset of household objects such as chair, coffee mug, door etc. which is of greater importance for visually impaired people. We have collected 773 training and 663 testing images, the images are captured with changes such as illumination, scale, viewpoint, rotation and background. All the images captured by iball CHD20 low cost CMOS camera from 0.5-3 meter distance. Some of the images from each category are depicted in figure 3.1, from top to bottom row the objects are banana, bin, bottle, calculator, chair, coffee mug, door, keyboard, lock, mobile, orange, shoes, sleeper, stairs.

MIT Indoor sub dataset: In second experiment we select 20 most common categories of indoor environment from MIT Indoor dataset [36] which are of greater importance for visually impaired people, in each category 80 training and 20 testing images are taken, so total 2000 images are used for this experiment. Evaluation of algorithm on this dataset is done because contextual information is very helpful for

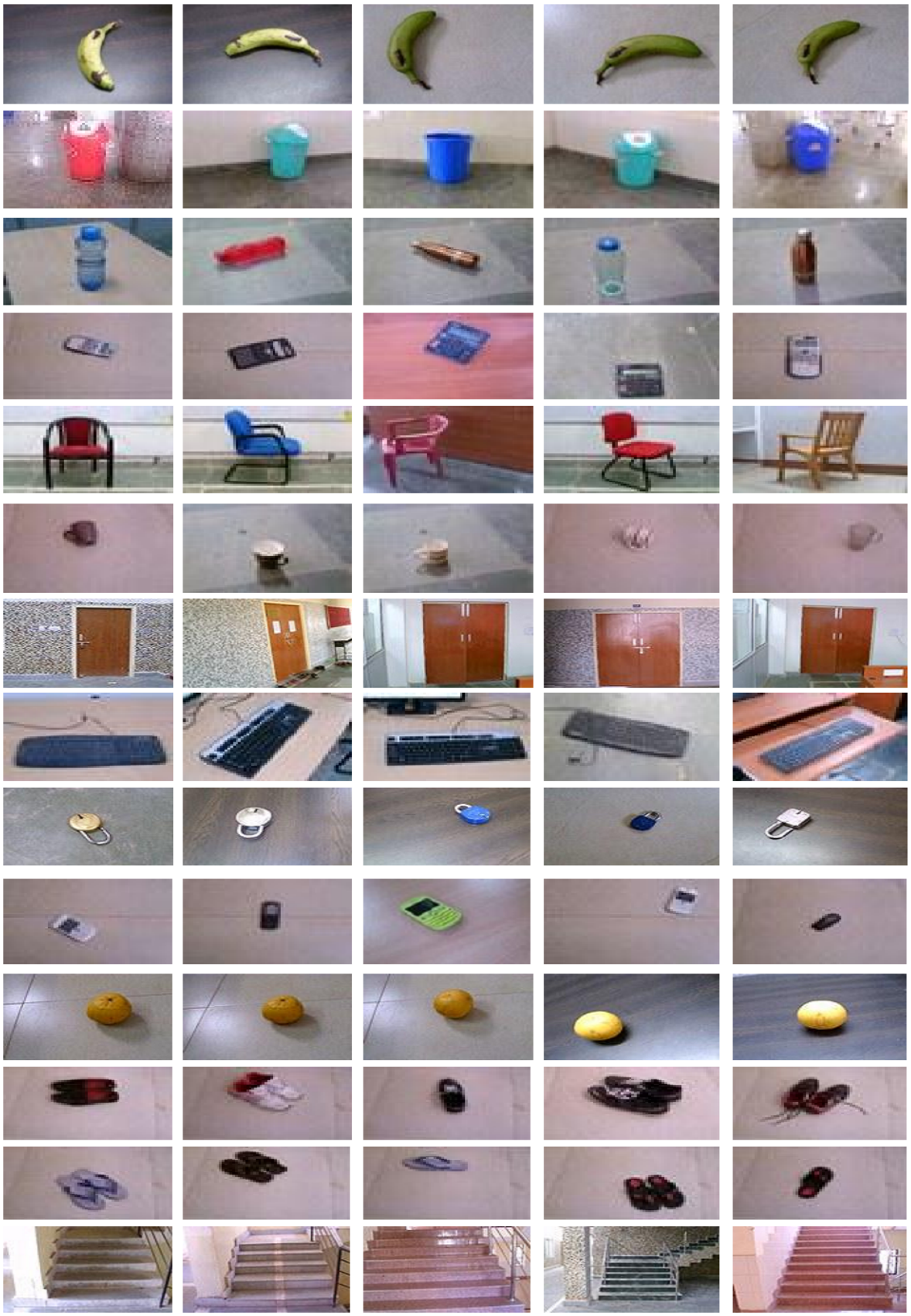


Figure 3.1: Our dataset(Household dataset).

blind person, and particularly in unfamiliar environment.

3.2 Classifiers

In order to classify captured image into one of the classes, classifier is required, it could be as simple as minimum distance classifier or as complex as neural network or support vector machine. In previous chapter feature extraction is discussed, in this section KNN and Softmax classifiers are explained. For object recognition, extracted features are used to train classifier parameters, to perform the classification and evaluation its performance, the dataset need to be divided into three parts: training, cross validation and testing.

3.2.1 Nearest Neighborhood

Nearest neighborhood classifier is an effective and simple way to classify the query image into one of the predefined classes. In the figure 3.3 a two dimensional feature vector is shown, the samples $*$, $+$ are belongs to two classes, now if some unknown sample o needed to assign some category label where it belongs to. To classify it into one of the classes, K number of nearest examples could be taken, K value could be 1,3,5 etc. For $K = 1$ the query sample is classified on the basis of one nearest neighbor so in this case o is classified as class-2 but for $k = 3$ the query sample is classified on the basis of majority votes among 3 nearest neighbors and hence o is classified into class-1.

Nearest neighbor can be searched(3.1) using Euclidean distance between query and training samples. $d(x^i, x^j)$ is the distance between sample x^i and x^j.

$$d(x^i, x^j) = \sqrt{\sum_{r=1}^{n} (x_r^i - x_r^j)^2} \tag{3.1}$$

where $x \in R^n$

Euclidean distance is not the only distance metric, one can use other distances

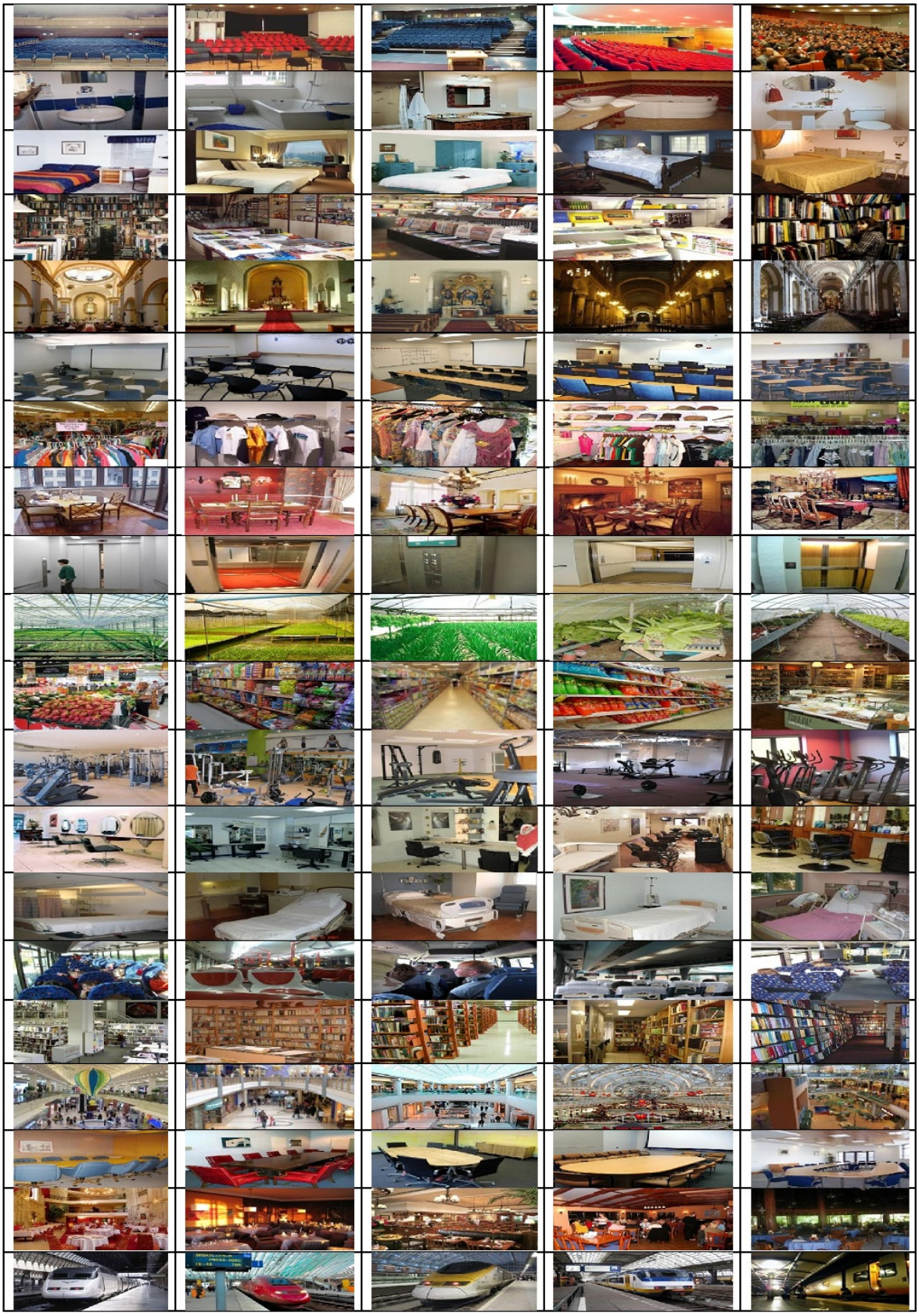


Figure 3.2: MIT Indoor sub dataset.

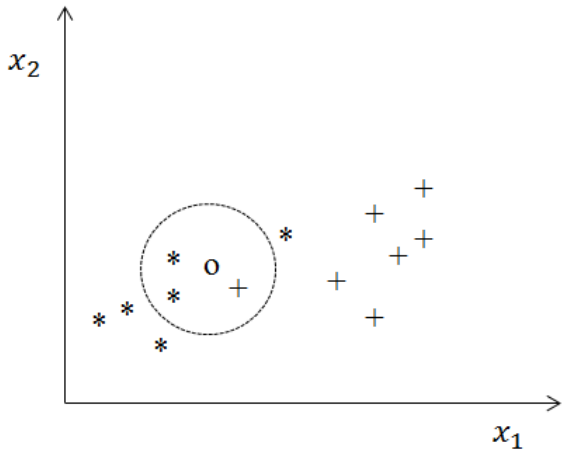


Figure 3.3: Two dimensional feature space.

such as chebyshev, Minkowski etc.

Minkowski distance metric:

$$d(x^i, x^j) = \sqrt[P]{\sum_{r=1}^{n} |x_r^i - x_r^j|^P} \tag{3.2}$$

Chebychev distance metric:

$$d(x^i, x^j) = max_r(|x_r^i - x_r^j|) \tag{3.3}$$

Chebychev distance metric is a special case of Minkowaski, when P tends to infinite.

Correlation distance metric:

$$d(x^i, x^j) = 1 - \frac{(x^i - \bar{x^i})(x^j - \bar{x^j})'}{\sqrt{(x^i - \bar{x^i})(x^i - \bar{x^i})'}\sqrt{(x^j - \bar{x^j})(x^j - \bar{x^j})'}} \tag{3.4}$$

where $\bar{x^j} = \frac{1}{m}\sum_r x_r^j$ and $\bar{x^i} = \frac{1}{m}\sum_r x_r^i$.

3.2.2 Softmax Classifier

Softmax classifier is the generalization of logistic regression[38]. In this section, first logistic regression will be explained and then it will be extended to multiclass classification using softmax function, which is nothing but softmax classifier.

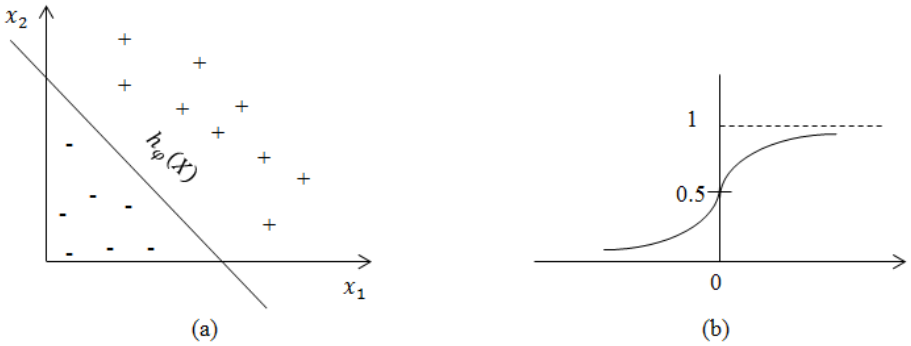


Figure 3.4: (a) Linearly separable data samples, (b) sigmoid function.

The name logistic regression came from its hypothesis function which is nothing but Logit function. Logistic regression is a popular classifier, in figure 3.4(a) we have depicted linearly separable case of binary class classification, in this particular case linear boundary can classify the samples into two classes. For this case the hypothesis could be as follows:

$$h_\phi(X) = f(\phi_0 + \phi_1 x_1 + \phi_2 x_2) \tag{3.5}$$

where ϕ_0, ϕ_1, ϕ_2 are free parameters need to be trained and x_1, x_2 are features extracted from the data.

For logistic regression the logit function is as follows:

$$f(\theta) = \frac{1}{1 + e^{-\theta}} \tag{3.6}$$

Logit function is shown in figure 3.4(b).

The hypothesis can be written as in (3.7).

$$h_\phi(x) = p(y = 1|x; \phi) \tag{3.7}$$

The equation shows probability that $y = 1$ for given x, and parameterized by ϕ. Note that for binary class classification

$$p(y = 0|x; \phi) + p(y = 1|x; \phi) = 1 \tag{3.8}$$

Now we need to train these free parameters, for that loss function or cost function is required as in (3.9).

$$L(\phi) = \frac{1}{m} \sum_{i=1}^{m} cost(h_\phi(x_i) - y_i) \tag{3.9}$$

The cost function for logistic regression is as below:

$$cost(h_\phi(x) - y) = \begin{cases} -log(h_\phi(x)) & \text{if } y = 1 \\ -log(1 - h_\phi(x)) & \text{if } y = 0 \end{cases}$$

From above equations loss function can be rewritten as follows:

$$L(\phi) = \frac{-1}{m} \sum_{i=1}^{m} y^i log(h_\phi(x^i)) + (1 - y^i) log(1 - h_\phi(x^i)) \tag{3.10}$$

Now to fit the parameter ϕ following minimization is required

$$\underset{\phi}{\operatorname{argmin}}\, L(\phi) \tag{3.11}$$

The hypothesis to predict class for query sample is as in (3.12).

$$h_\phi(x) = \frac{1}{1 + e^{-\phi^T x}} \tag{3.12}$$

The minimization is done by using gradient decent algorithm, which is simple and effective algorithm to optimize cost function . The gradient decent algorithms is shown below:

Repeat:

{

$$\phi_j = \phi_j - \alpha \sum_{i=1}^{m} (h_\phi(x^i) - y^i) x_j^i \tag{3.13}$$

}

where

$$\frac{\partial L(\phi)}{\partial \phi} = (h_\phi(x^i) - y^i) x_j^i \tag{3.14}$$

The sigmoid function has a very useful property, that is, its derivative can be easily expressed in terms of its output.

The concept of logistic regression can be further extended to multi-class classification using the softmax function(3.15).

$$p(y = j|x; \phi) = \frac{e^{\phi_j^T x}}{\sum_{j=1}^{K} e^{\phi_j^T x}} \tag{3.15}$$

the hypothesis function is as follows

$$h_\phi(x) = \frac{1}{\sum_{j=1}^{K} e^{\phi_j^T x}} \begin{bmatrix} e^{\phi_1^T x} \\ e^{\phi_2^T x} \\ . \\ . \\ . \\ e^{\phi_K^T x} \end{bmatrix} \tag{3.16}$$

The modified cost function is as follows:

$$L(\phi) = \frac{1}{M}\left[\sum_{i=1}^{M}\sum_{j=1}^{K} t_{ij} log\left(\frac{e^{\phi_j^T x}}{\sum_{l=1}^{K} e^{\phi_l^T x}}\right)\right] \tag{3.17}$$

Where t_{ij} is one if and only if sample belongs to target class, otherwise zero.

3.3 Proposed Object Recognition Method

In this section we have explained proposed object recognition algorithm, the block diagram of our method is shown in figure 3.5, we are using convolutional neural network and recursive neural network in one pipeline, and Gabor feature extraction followed by pooling stage (similar as in convolutional neural network) which goes to recursive neural network in second pipeline, the combined features vector is used to train softmax classifier.

CNN stage is tarined using unsupervised learning in similar way as described in [10]. We train K filters of size 9×9, which are convolved with the input image of size 148×148 as in (3.18), the resulting response is 140×140 dimensional K output maps. These maps are further passed though rectified linear unit [9] and contrast normalization [37] stages respectively as in equation (3.19) and (3.20). Now these normalized map goes to average pooling stage as in equation (3.21), the output of pooling stage is maps of 27×27 dimension.

$$v_j^n = \sum_{k=1}^{C} w_{kj}^n * v_k^{n-1} \tag{3.18}$$

$$v_j^{n+1} = max(0, v_j^n) \tag{3.19}$$

Where v_k^{n-1} is k^{th} input map, v_j^n is j^{th} output map and w_{kj} is trained filter, $j = 0, 1, 2, 3, ...K$, C is 3 for color images.

$$v_j^{n+3}(p,q) = \frac{v_j^{n+2}(p,q)}{max(\epsilon, \sigma^{n+1}(\mathcal{N}(p,q)))} \tag{3.20}$$

where $\sigma^{n+1}(\mathcal{N}(p,q)) = \sqrt{\sum_{k,\bar{p}\in\mathcal{N}(p),\bar{q}\in\mathcal{N}(q)} w_{\bar{p}\bar{q}}.(v_k^{n+2}(\bar{p},\bar{q}))^2}$ and ϵ is a small number.

$$v_j^{n+3} = \frac{1}{N}\left(\sum_{\bar{p}\in\mathcal{N}(p),\bar{q}\in\mathcal{N}(q)} v_j^{n+2}\right) \tag{3.21}$$

CNN stage is followed by recursive neural network layer, recursive neural network is used to extract hierarchical feature representation by repeating same neural network recursively in a tree structure [12]. Each recursive neural network layer consists of three sub layers and any number of such RNNs can be used, for e.g. 64 CNN and 64 RNN generate 4096 dimentional feature vector.

The second pipeline consists of Gabor feature extraction stage , pooling(same as in (3.21)), and RNN on grayscale images, the Gabor filters (2.28) are convolved with the input image and produce M(scale×orientation) number of output maps, these maps are further processed by pooling stage witch is used to reduce the size and provide feature invariance to scale and shift, these reduced dimensional maps are processed by RNN stage to produce feature vector, this feature vector is augmented with the features from CRNN pipeline to generate final feature vector.

This final feature vector is used to train Softmax classifier. Softmax classifier is the generalization of logistic regression for multi-class classification, equation (3.16) and (3.17) are Softmax classifier hypothesis and cost function respectively.

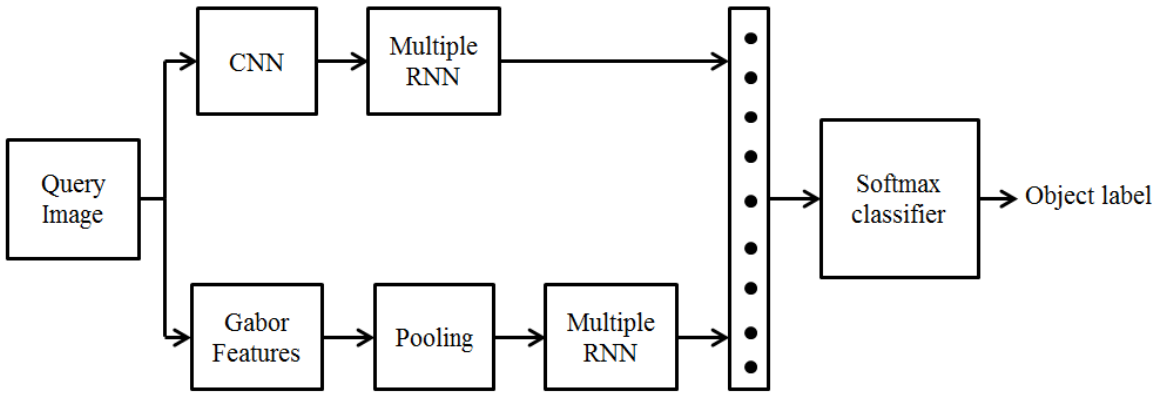


Figure 3.5: Proposed object recognition block diagram.

3.4 Results

In this section analysis and accuracy results are discussed, two experiments have been carried out, in first experiment, algorithm is tested for its performance on our household object recognition dataset, and in second experiment the algorithm is evaluated on MIT Indoor scene dataset to provide contextual information of the environment, GCRNN(proposed) comparison with other object recognition(unsupervised) methods and analysis is given. We are using Intel core i5-2400 CPU at 3.10 GHz with 4GB RAM Dell desktop for simulation results; iball CHD20 CMOS camera is used to capture images. Experiments are performed and simulation results are depicted using MATLAB verson R2012b. Apart from this we cross validate the algorithm on standalone low cost computing device(Raspberry Pi [39]) using OpenCV library [40].

Object category classification accuracy formula given in equation 3.22.

$$Accuracy = \frac{Corectly\ \ classified\ \ objects}{Total\ \ number\ \ of\ \ objects} \times 100 \tag{3.22}$$

In table3.1 proposed algorithm is compared with Kmean1600 [10] and CRNN [12] unsupervised learning methods for object recognition. GCRNN is using 64CNN, 32RNN in first pipeline, and 40 Gabor maps, 128RNN in second pipeline (total 7,168 features)though out the experiment, mentioned otherwise.

In the table category accuracy as well as average accuracy is given. Proposed

Category	Kmean1600	CRNN	GCRNN(Proposed)
Banana	84.00	92.00	**92.00**
Bin	84.13	90.48	**96.83**
Bottle	82.14	86.67	**86.67**
Calculator	**100**	71.43	72.73
Chair	96.40	97.30	**99.10**
Coffee mug	100	100	**100**
Door	96.71	96.05	**97.37**
Keyboard	34.78	73.90	**82.61**
Lock	92.31	87.18	**97.44**
Mobile Phone	66.67	66.67	**66.67**
Orange	75.00	95.00	**95.00**
Shoes	97.75	**100**	98.88
Sleeper	85.00	90.00	**90.00**
Stairs	93.33	93.33	**93.33**
Average accuracy (%)	84.87	88.57	**90.62**

Table 3.1: Category recognition accuracy(%) on our dataset.

algorithm outperform both the methods on almost all categories and clearly outperform in terms of average accuracy.

Second experiment is done using MIT Indoor scene sub dataset for contextual information, as depicted in table3.2 our object recognition algorithm outperform CRNN [12] with 8192 as well as 16384 (CRNN with its highest accuracy) features. In this experiment our feature vector size is 7168.

In table 3.3 an exhaustive analysis is carried out for GCRNN algorithm. In this experiment 40 Gabor filters(5 Scale and 8 Orientation) are fixed while CNN and RNN of first pipeline have been observed. From the results it is clear that the accuracy is not much deteriorated even when we use very less number of CNN and RNNs.

In figure 3.6 it is depicted that CRNN on color images and GRNN on grayscale

Category	CRNN(8192)	CRNN(16384)	GCRNN(Proposed)
Auditorium	60	55	**55**
Bathroom	35	40	**75**
Bedroom	40	**55**	35
Book store	20	20	**20**
Church inside	65	65	**75**
Class room	55	55	**60**
Clothing store	**40**	30	30
Dining room	45	35	**70**
Elevator	60	70	**70**
Greenhouse	90	95	**95**
Grocery store	**65**	50	40
Gym	45	35	**45**
Hair salon	25	35	**35**
Hospital room	15	15	**15**
Inside bus	75	70	**85**
Library	20	15	**25**
Mall	35	45	**45**
Meeting room	30	**45**	40
Restaurant	15	15	**35**
Train station	70	75	**75**
Average accuracy (%)	45.25	46.00	**51.25**

Table 3.2: Category recognition accuracy(%) on MIT Indoor dataset.

Category	CNN8	CNN16	CNN32	CNN64	CNN128	CNN256
RNN4	90.90	91.21	91.52	92.49	92.03	92.18
RNN8	90.95	91.29	91.97	92.53	92.62	91.77
RNN16	91.05	91.58	92.29	93.00	93.06	92.96
RNN32	91.21	91.86	92.53	93.25	92.79	92.75
RNN64	91.34	92.24	92.82	93.15	92.77	91.92
RNN128	91.46	92.45	92.78	93.15	92.25	91.41
RNN256	91.83	92.74	92.68	92.27	91.88	90.95

Table 3.3: GCRNN analysis, accuracy (%).

images are extracted different features from the images and when we combine these features the accuracy improve with marginal difference.

In figure 3.6 bar plot analysis of GCRNN is portrayed, as stated earlier we combine convolutional neural network(CNN) and recursive neural network(RNN) pipeline with Gabor and recursive neural network(GRNN) pipeline, which gives us higher accuracy than best accuracy achieved by Kmean1600 and CRNN. We are using very less number of features from CRNN(2048) and combine with GRNN features(5120) to get a feature vector of size 7168, and as shown in figure 3.6 combined feature outperform both of them(CRNN and GRNN) with 10% improvement. GRNN features analysis with respect to number of RNN is portrayed in figure 3.8, after 64 RNN the improvement in accuracy is not significant. In this experiment we are using 40 Gabor feature maps.

In table 3.4 Softmax and KNN classifier accuracy is compared, Softmax classifier clearly outperforming KNN classier, but for two categories likewise lock and mobile phone KNN is giving better results. This table is demonstrating that even simple classifier like KNN can work reasonably well with our feature extractor stage. We are using correlation distance metric and exhaustive search method based KNN, the reasoning behind this selection is demonstrated with the help of table 4.1.

As shown in table 3.5 correlation distance metric and Minkowski using exhaustive search, and Minkowski using kd tree search with one neighbor are giving highest ac-

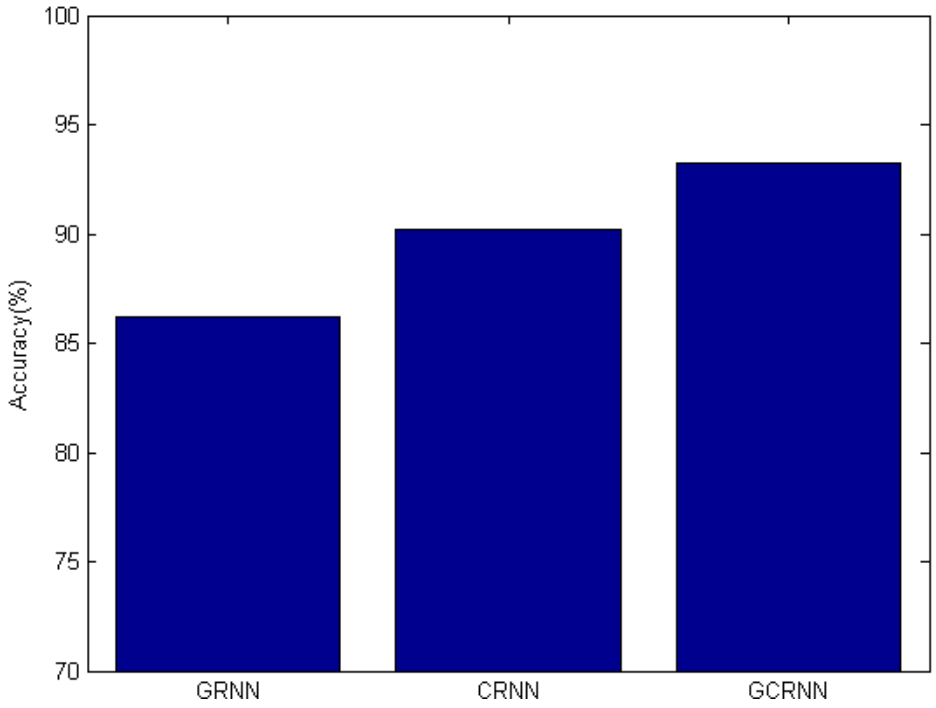


Figure 3.6: CRNN, GRNN and GCRNN on Household objects dataset.

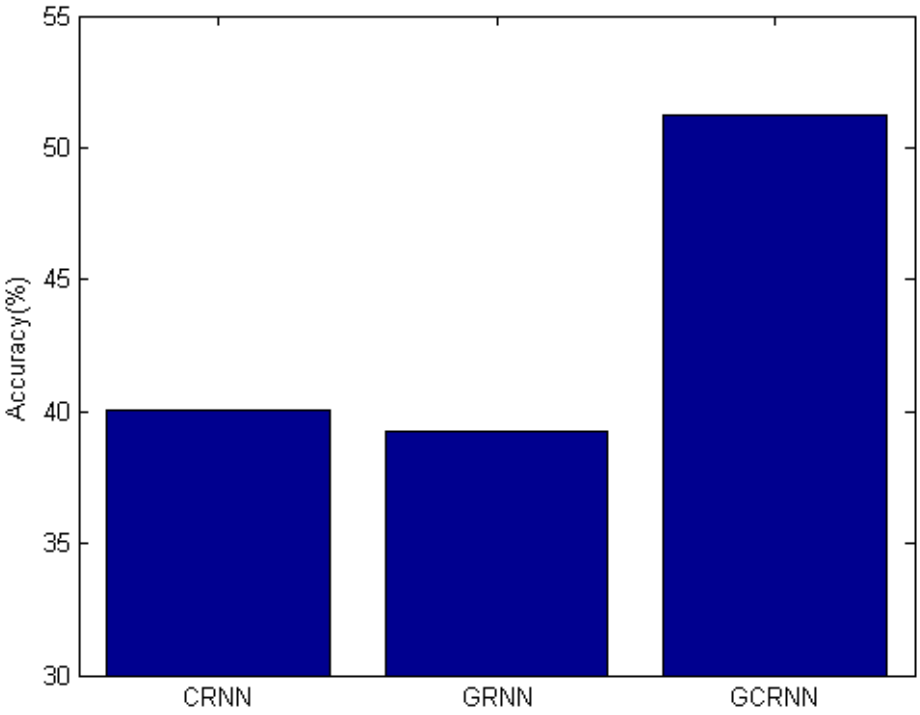


Figure 3.7: CRNN, GRNN and GCRNN on MIT Indoor sub dataset.

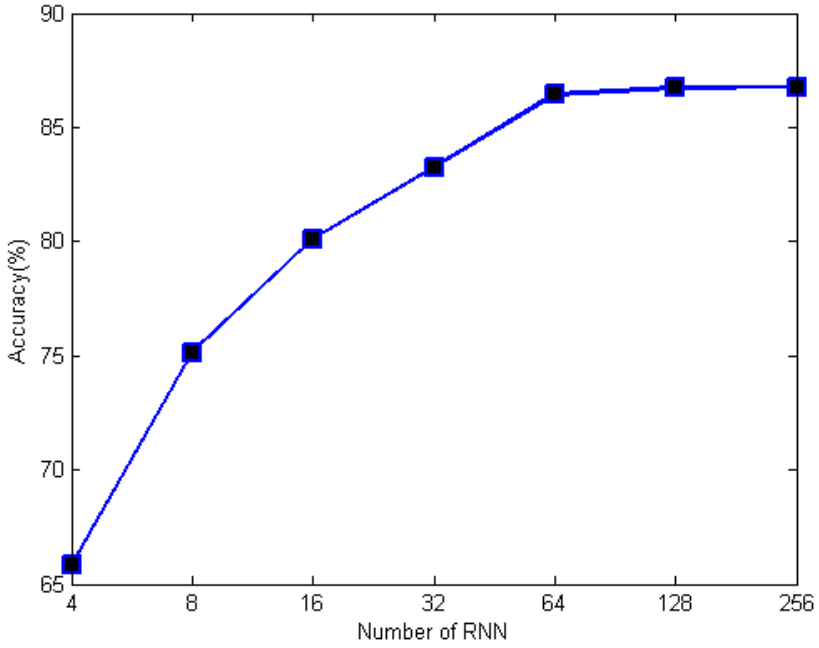


Figure 3.8: GRNN accuracy(%) with respect to number of RNN on our dataset.

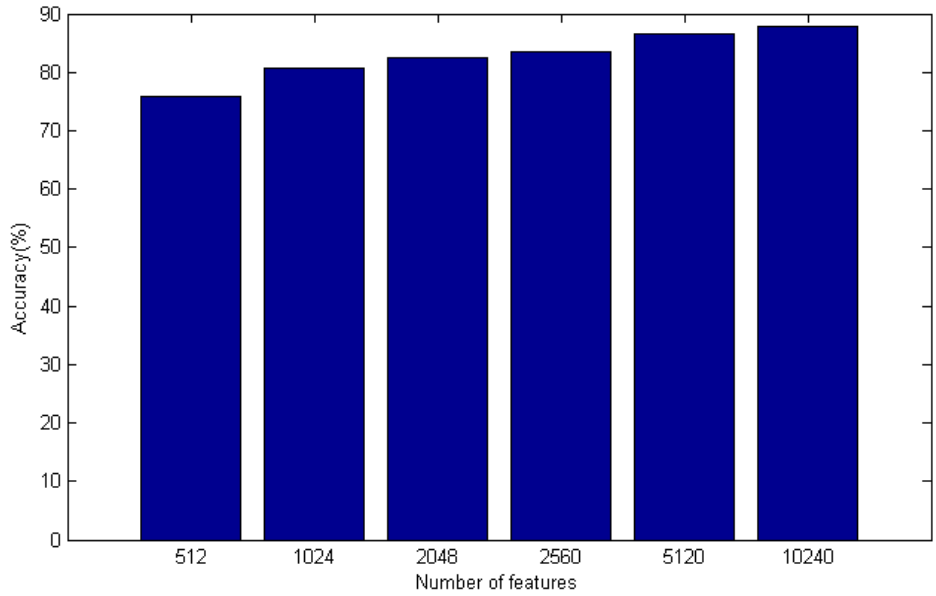


Figure 3.9: GRNN accuracy(%) with respect to Gabor features variations and 128RNN.

Category	GCRNN with KNN	GCRNN with Softmax
Banana	84.00	**92.00**
Bin	79.37	**96.83**
Bottle	85.71	**86.67**
Calculator	61.90	**72.73**
Chair	94.59	**99.10**
Coffee mug	100	**100**
Door	92.10	**97.37**
Keyboard	82.60	**82.61**
Lock	**100**	97.44
Mobile Phone	**76.19**	66.67
Orange	95.00	**95.00**
Shoes	94.38	**98.88**
Sleeper	75.00	**90.00**
Stairs	86.66	**93.33**
Average accuracy (%)	86.25	**90.62**

Table 3.4: GCRNN comparison with KNN and Softmax classifier.

curacy. But the prediction of KNN classifier is very much faster with correlation as compare to Minkowski distance metric. Due to this reason we are using correlation distance metric. In figure 3.10, GCRNN is evaluated for repeatability, accuracy with respect to number of times it run is plotted, it can be noticed that the repeatability is very high. The standard deviation is 0.0041. High repeatability ensure that system will produce reported results every time we run it.

3.5 Summery

In this chapter we introduce our dataset, and object recognition algorithm. Our object recognition method is able to produce better accuracy with less number of features as shown in results. We have depicted analysis of GRNN with respect to number of

Search Method	Distance Metric	Neighbors	Accuracy in %
Exhaustive	Correlation	1	**89.00**
	Euclidean	1	87.96
	Minkowski(P=4)	1	**89.00**
	Chebychev	1	77.26
	Correlation	3	87.81
	Euclidean	3	86.62
	Minkowski(P=4)	3	87.37
	Chebychev	3	74.29
KD Tree	Euclidean	1	87.96
	Chebychev	1	77.26
	Minkowski(P=4)	1	**89.00**
	Euclidean	3	86.62
	Chebychev	3	74.29
	Minkowski(P=4)	3	87.37

Table 3.5: KNN classifier analysis.

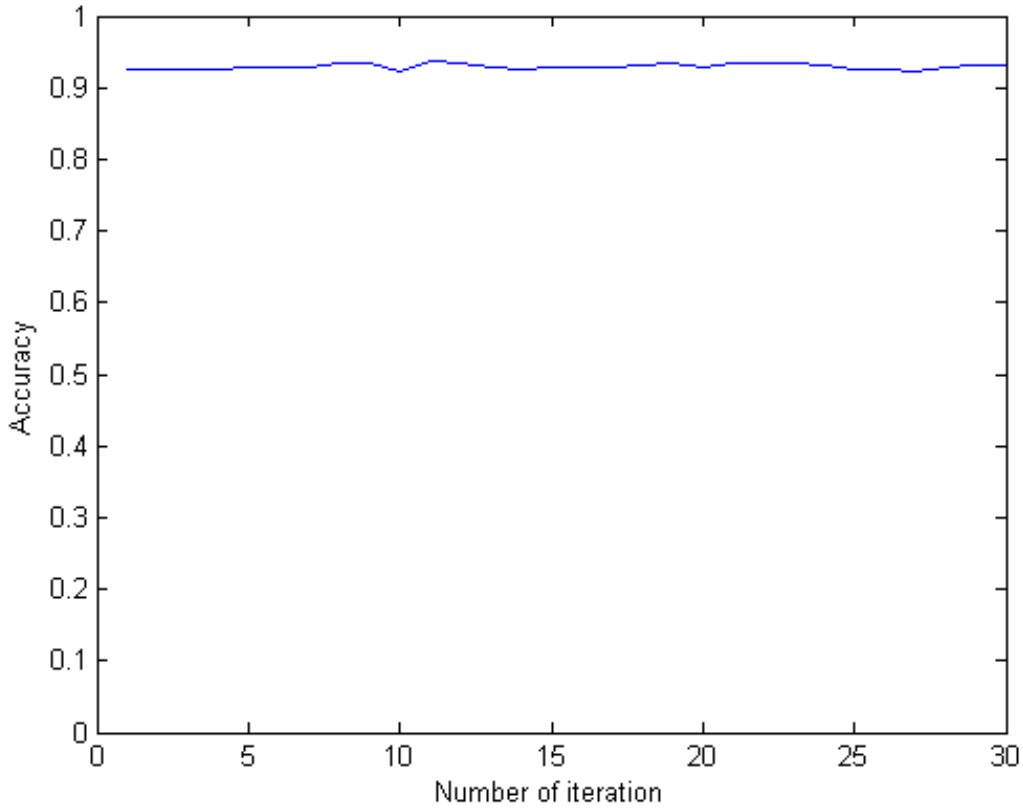


Figure 3.10: GCRNN accuracy with respect to number of iteration(repeatability).

recursive neural network. The comparison of GCRNN with KNN and Softmax classifier is depicted. GCRNN has high repeatability as depicted in figure 3.10.

Chapter 4

Assistive system

Technology is becoming more and more handy and useful every day, likewise several assistive systems have been developed so far such as Virtual white cane [22],vOICe [41] etc. Though these systems are very effective for navigation but for highly complex task like object recognition these systems does not work. Object recognition is crucial not only to manipulate objects but also for way finding and navigation. We are using object recognition along with color recognition to help visually impaired people.

4.1 Proposed System

The proposed assistive system consists of four main sub block as follows:

- Mode selection
- Color recognition
- Object recognition
- Speech synthesizer

The system block diagram is portrayed in Figure 4.1. Mode selection is a very useful system feature which allows user to switch the functionality of the system and

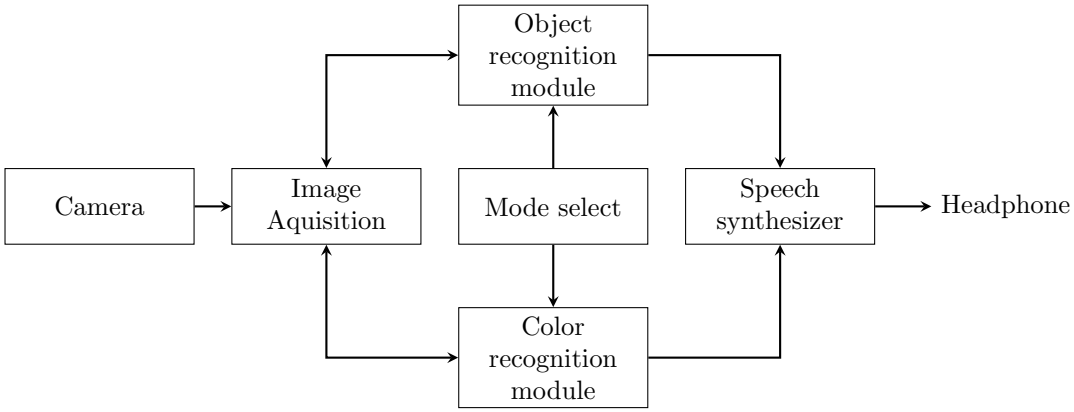


Figure 4.1: Proposed system block diagram.

exploit on demand feature that means the user has accessibility of color and object recognition when required only, it is due the reason that we do not want to affect the auditory capability of blind person, because blind or visually impaired people rely very much on there auditory capability. The output of mode selection block goes to one of the image processing blocks, either color, or object recognition, the output of the selected block is processed by speech synthesizer and produce sound as feedback to the user. Each sub block is described in the following sub sections.

In figure 4.2 proposed system flow chart is depicted, first of all system scan general purpose input output(GPIO), if Push button 1(PB1) is pressed by user then system will capture image in front of camera and process it for color recognition, the color recognition algorithm is described in section 4.1.2, the color label generated by this stage goes to *espeak* speech synthesizer. On the other hand if Push button (PB2) is pressed then image is captured and processed for object recognition, proposed algorithm for object recognition is described in section 3.3. The output label from object recognition stage goes to speech synthesizer. The whole process repeated again and again in an infinite loop.

4.1.1 User Interface

User interface panel consists of two on demand Push buttons. Push button panel circuit diagram is shown in figure 4.3, the circuit contains two $10Kohm$ pull-up resistors and

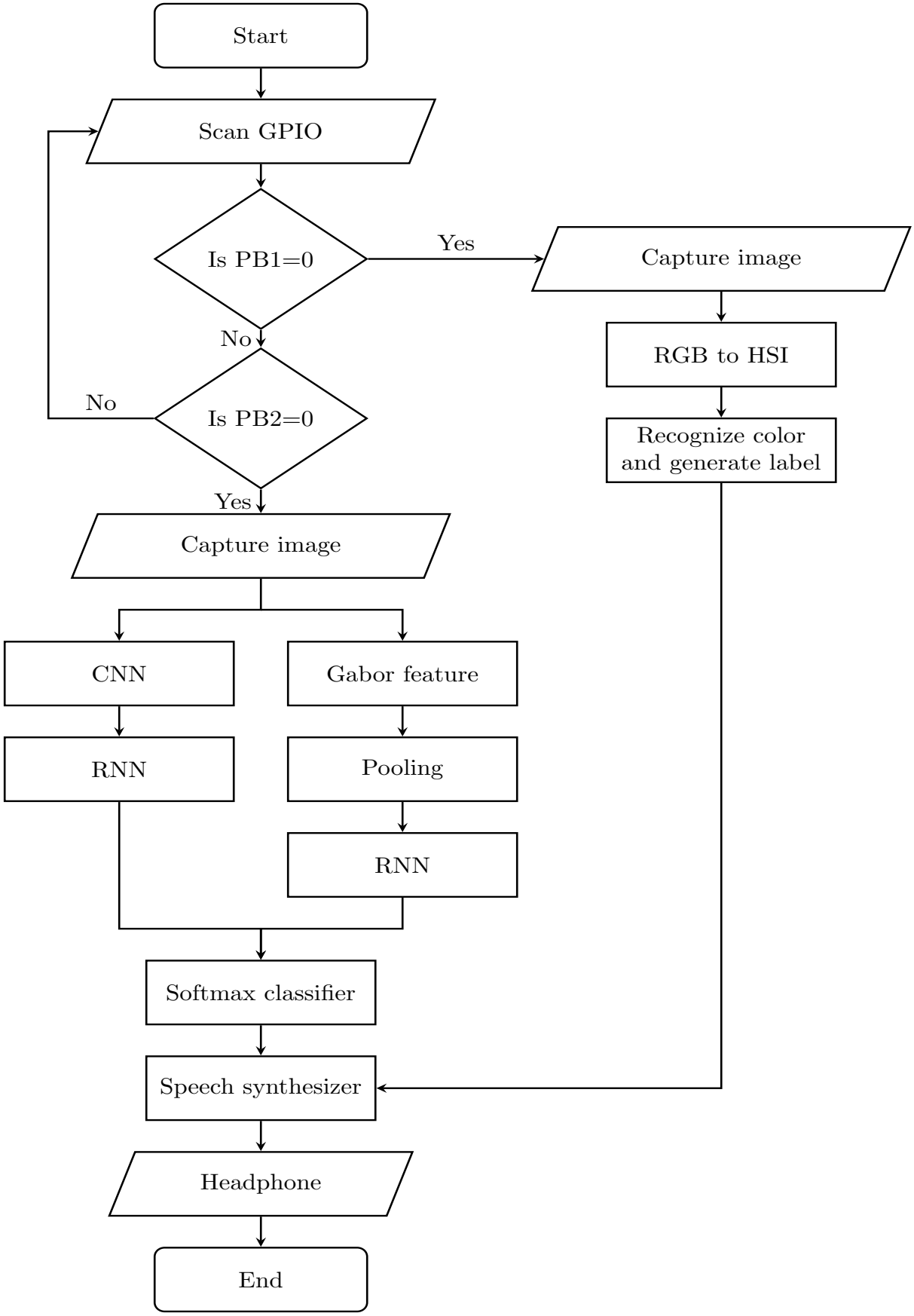


Figure 4.2: System flow chart.

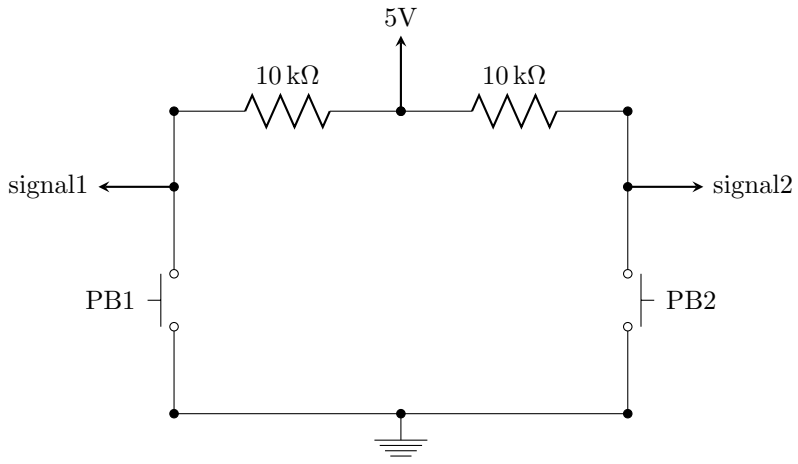


Figure 4.3: User interface circuit.

two push buttons PB1 and PB2, *signal*1 and *signal*2 is used to read PB1 and PB2 status. When push button PB1 is pressed signal1 goes to low state and remain in high state if PB1 is not pressed, same for PB2. Push button status can be read by GPIO pins of Raspberry pi board.

4.1.2 Color Recognition Module

Color recognition module found its importance in particular for low vision people and for those who have lost their sight in some accident; it is also useful to provide additional information, because every object carry some color attribute along with it.

We are using HSI (Hue, Saturation, Intensity) color space to recognize colors, it is due to the fact that this color space is most similar to the way human define the colors and intensity invariance of Hue. Due to these reasons, we first transform RGB (Red, Green, Blue) color space to HSI color space and process it further. Now to recognize the color of object in front of the camera we first separate each channel in HSI color space and then select a small window of size 20×20 around the center pixel of the image in each channel. In this sub-image we sample pixels randomly, and on the basis of cardinality of majority pixels, system decide the color of object using observed threshold and produce color label as its output. We analyze several cameras in our Lab such as Logitech C110, Logitech C120, iball CHD20 etc. for Hue, Saturation and

Color	Hue	Saturation
Red	0.94-1.0	0.4-1.0
Green	0.35-0.45	0.4-1.0
Blue	0.55-0.65	0.67-1.0
Yellow	0.15-0.2	0.55-1.0
Cyan	0.55-0.65	0.5-0.65
Purple	0.66-0.75	0.4-0.6
Brown 1	0.88-0.98	0.0-0.4
Brown 2	0.0-0.14	0.4-0.55
Brown 3	0.0-0.14	0.55-0.65

Table 4.1: Color thresholding table.

Intensity, and we make a thresholding table 4.1.

The RGB to HSV conversion is done using the equations given below, equation 4.1 is used to get intensity value, and on the basis of value V, Saturation and Hue is computed as in equation 4.2 and 4.3. In algorithm 1, x, y are random coordinates; *hp*, *sp*, *vp* are hue, saturation and value patch respectively; 100 pixels are randomly sampled from a 20×20 window and the test pixels for conditions.

$$V = max(R, G, B) \tag{4.1}$$

$$S = \begin{cases} (V - min(R, G, B))/V & \text{if } V \neq 0 \quad (4.2a) \\ 0 & \text{if } V = 0 \quad (4.2b) \end{cases}$$

$$H = \begin{cases} 60((G - B)/(V - min(R, G, B))) & \text{if } V = R \quad (4.3a) \\ 60(2 + ((B - R)/(V - min(R, G, B)))) & \text{if } V = G \quad (4.3b) \\ 60(4 + ((R - G)/(V - min(R, G, B)))) & \text{if } V = B \quad (4.3c) \end{cases}$$

Algorithm 1 Color recognition algorithm

$x \leftarrow random(0 \text{ to } 20,10)$

$y \leftarrow random(0 \text{ to } 20,10)$

$hsv \leftarrow$ convert RGB to HSV

$hp \leftarrow h(H/2-10:H/2+10, W/2-10:W/2+10)$

$sp \leftarrow s(H/2-10:H/2+10, W/2-10:W/2+10)$

$vp \leftarrow v(H/2-10:H/2+10, W/2-10:W/2+10)$

$rct, gct, bct, yct, blkct, wct \leftarrow 0$

$t \leftarrow 0$

for $t \leftarrow 0:10$ **do**

    if $sp(x[t], y[t]) \geq 0.2$ **then**

        if $hp(x[t], y[t]) \geq 330$ **then**

            $rct \leftarrow r+1$

        else if $(hp(x[t], y[t]) \geq 140) and (hp(x[t], y[t]) \leq 180)$ **then**

            $gct \leftarrow gct+1$

        else if $(hp(x[t], y[t]) \geq 210) and (hp(x[t], y[t]) \leq 250)$ **then**

            $bct \leftarrow bct+1$

        else if $(hp(x[t], y[t]) \geq 5) and (hp(x[t], y[t]) \leq 40)$ **then**

            $yct \leftarrow yct+1$

        else

            color not define

        end if

    else

        if $(vp(x[t], y[t]) \geq 0.7)$ **then**

            $wct \leftarrow wct+1$

        else if $(vp(x[t], y[t]) \leq 0.2)$ **then**

            $blkct \leftarrow blkct+1$

        else

            $gray$

        end if

    end if

end for

$index \leftarrow argmax([rct, gcr, bct, yct, blkct, wct])$

$Label \leftarrow colordictionary(index)$

4.1.3 Object Recognition Module

Object recognition module is used to recognize nearby objects such as mobile phone, coffee mug, sleeper etc. we are using our object recognition algorithm which we have described in chapter 3. The implementation results are given in next section.

4.1.4 Speech Synthesizer

For speech synthesizer we are using espeack module of Raspberry Pi, espeak is an open source speech synthesizer for linux and windows. The syntax for espeak is $espeak[option][words]$.

Main features of espeak are as follows:

- Different voices are includes and there characteristics can be changed.
- written in C++.
- support several options.

Some useful options:

- -a: Amplitude control from 0 to 200.100 by default.
- -g: gap between words, default value is 10ms.
- -p: pitch adjust between 0 to 99, default value is 50.
- -s: word per minute range is 80 to 370, default value is 170.
- –compile voice name: It compile the pronunciation rules,voice name is used to specify the language.
- -voices language: This option list the available voice for specified language.

4.2 Hardware Implementation

We are using Raspberry Pi, low cost Broadcom BCM2835 multimedia application processor based hardware, We choose this hardware due to its low power, small size and low cost at one hand and on the other hand the implementation done on this board can be replicated on any modern smart phone. Linux operating system is running on this platform which is installed on 16 GB SD card, for algorithm implementation we are using OpenCV image processing library and Python programming language.

4.2.1 User Interface Pannel

User interfacing panel top view is shown in figure 4.4, black and pink wires are for ground and 5 volts respectively, and two red wires next to pink wire are for signal. The panel interfacing with Raspberry Pi is as follows:

GPIO Pin 3 – Signal 1 (Red wire).

GPIO Pin 4 – GND(black wire).

GPIO Pin 5 – Signal 2 (Red wire).

GPIO Pin 6 – 5 Volt(Pink wire).

4.2.2 Raspberry pi

Raspberry Pi is a low cost embedded hardware, availabel in open market. It contains ARM11 ARMv6 architecture 700 MHz CPU and 512 MB RAM. It support embedded Linux Operating system. There are several useful features of this hardware which attract our attention such as 3.5mm audio jack, 2 USB ports, on board networking, general purpose input output(GPIO) pins, samll size and light weight. Raspberry Pi board along with push button panel is shown in figure 4.5.

Raspberry pi features: Some of the raspberry pi features are enlisted below [39].

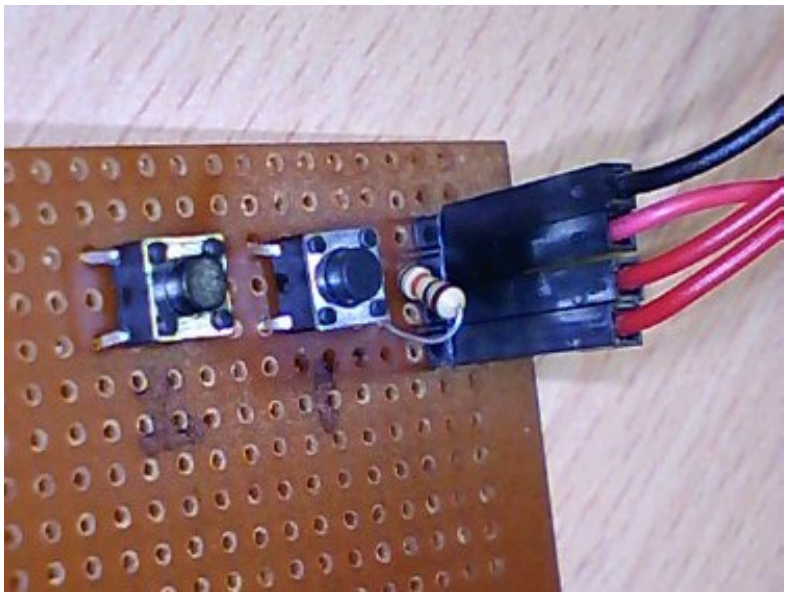


Figure 4.4: User interface panel.

- Broadcom BCM2835 SOC.
- 512 RAM SDRAM.
- 2 USB 2.0 Ports.
- One Ethernet Port.
- 1 HDMI Connector.
- 3.5 mm audio jack.
- 26 Pin GPIO port.
- SD card support.
- JTAG header.
- CSI camera connector.
- DSI Display connector.
- Micro USB connector for power.

Raspberry require Linux operating system for its operation, there are several customized Linux operating system available such as NOOBS, RASPBIAN, UBUNTU MATE etc. at official website of Raspberry Pi board.

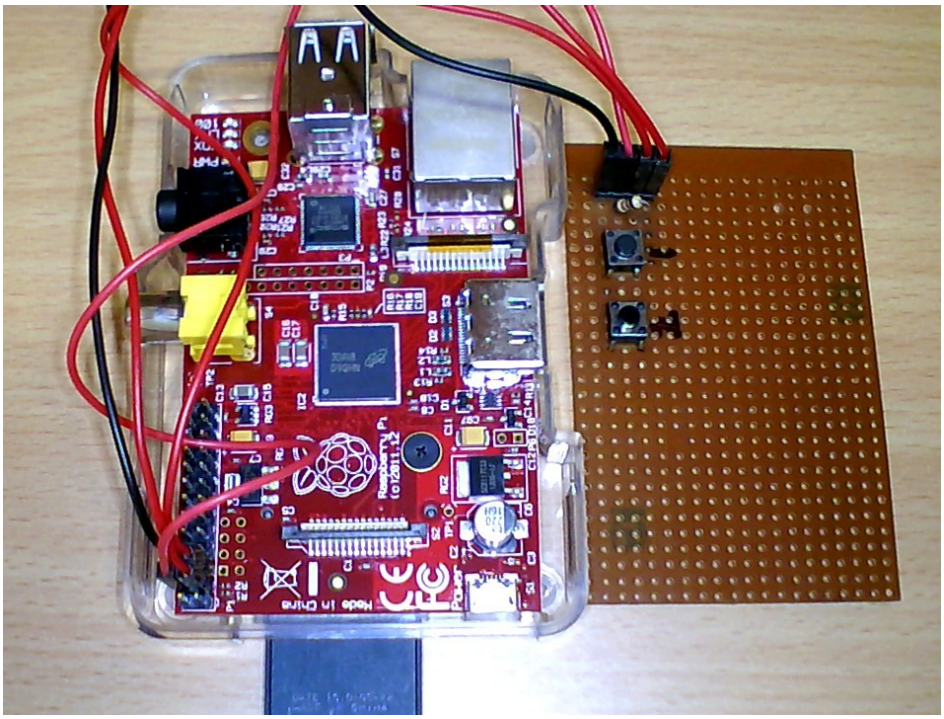


Figure 4.5: Raspberry pi board with interfacing panel.

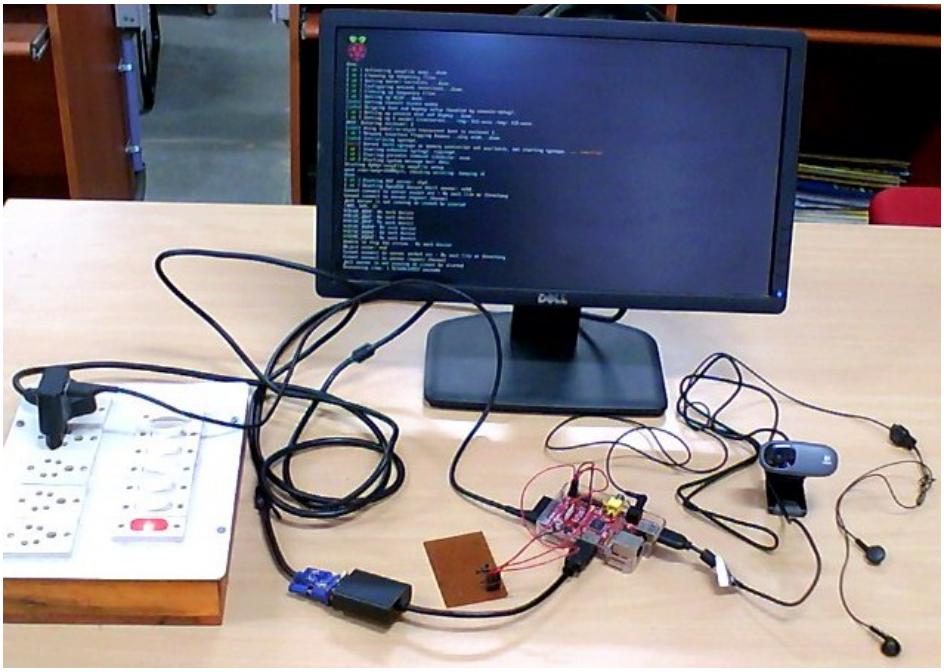


Figure 4.6: Raspberry pi board with interfacing panel and display.

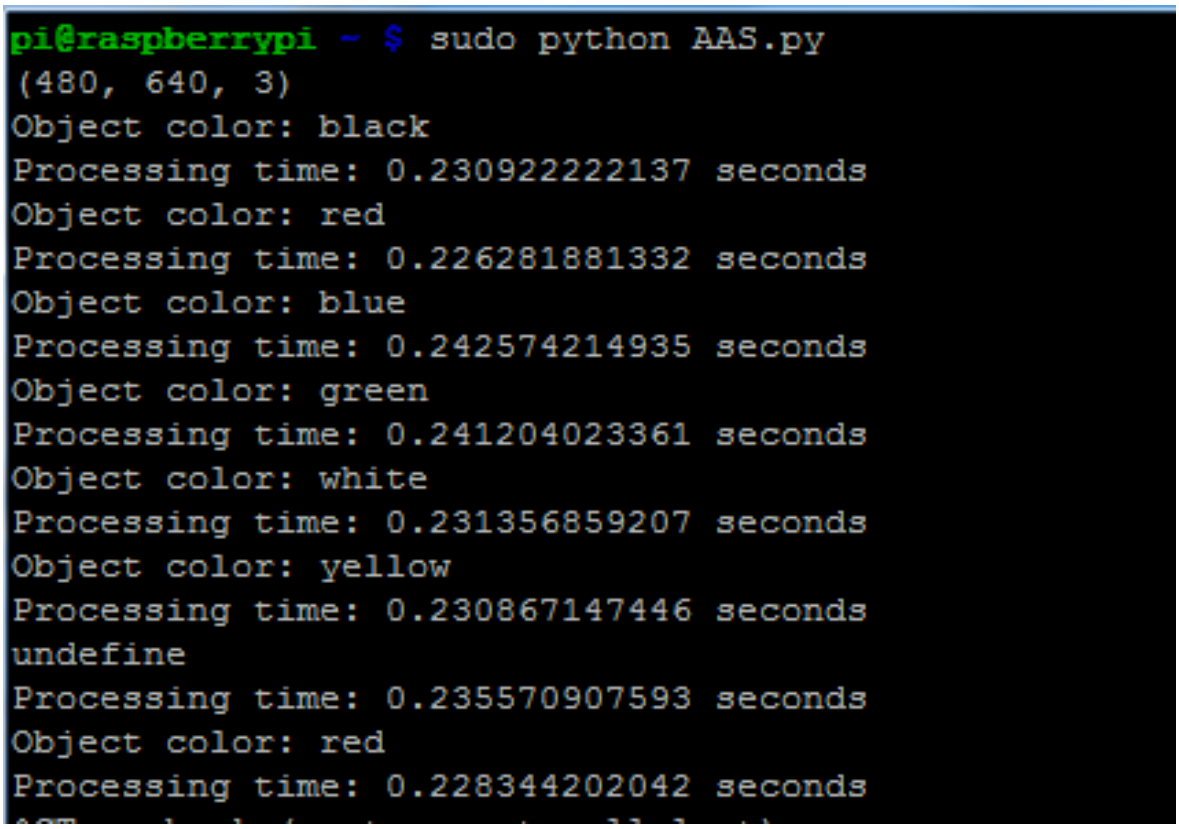


Figure 4.7: Raspberry pi terminal output of color recognition module.

Logitech C110 imaging camera is used to capture real world images, it is a 1.3 MP, CMOS, low power, USB camera. This can be connected to one of the two USB2.0 ports available on board. The implementation is done using OpenCV image processing Library and Python programming language.

4.3 Results

The hardware implementation is done using Raspberry Pi BCM 2835 system on chip board. Pi terminal output for color and object recognition modules are depicted in Figure 4.7 and 4.8. From terminal output figure 4.7 it can be noticed that our color recognition algorithm is taking around $230 milisecond$ to process $640 \times 480 \times 3$ image.

The category classification(label and Softmax classifier output) is depicted in figure 4.8, from the result it can be noticed that for category box, the probability in favor of box is 0.9998 which is apparently correct classification since box is first category for this

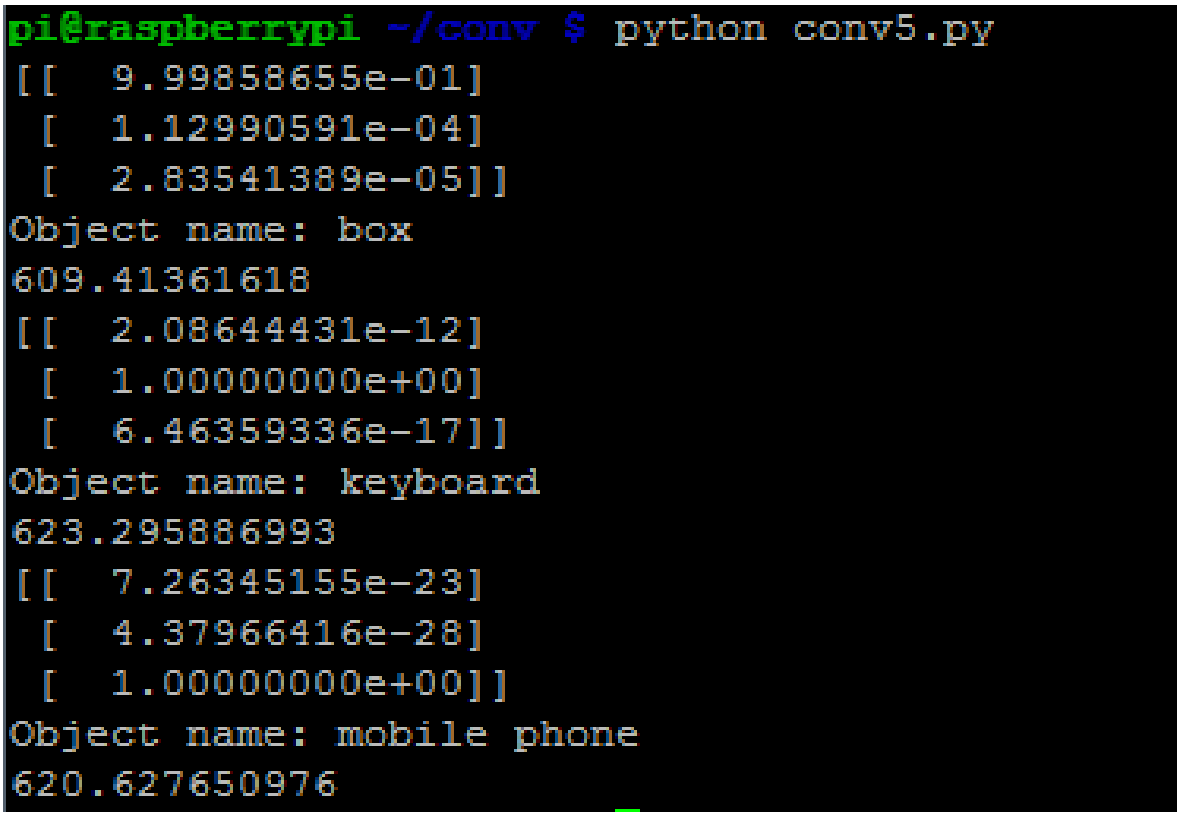


Figure 4.8: Raspberry pi terminal output of object recognition module.

dataset, similarly for second and third category (keyboard, mobile phone) classification the probability in favor of correct classification is 1.

4.4 Summery

In this chapter hardware and implementation details such as board used for image processing, operating system used, camera used are discussed. Color recognition algorithm introduced in section 4.1.2 along with it, observed Hue, Saturation and Intensity thresholding values are given here. Push button panel and its circuit is discussed well in section 4.1.1 and 4.2.1. Raspberry Pi board, its peripherals , connectors and operating system is discussed in section 4.2.2. Finally results of color and object recognition algorithm implementation are depicted in section 4.3.

Chapter 5

Conclusion

In this thesis we have proposed an assistive system for less privilege group of people in the society, which exploit object and explicit color recognition methods with on demand mode selection. GCRNN is found to be better in terms of accuracy with less number of features as compare to CRNN. Apart from simulation on MATLAB software, we have also done hardware implementation using embedded board, simulation and hardware implementations is done successfully. The processing time for color recognition module is very appropriate for this application. The GCRNN object recognition algorithm is highly parallel and hence the processing time can be improved using hardware such as FPGA, DSP, GPU etc. Generic object recognition to aid visually impaired people is in its very early stage of development, and has a very bright future; it is a great service for humanity to develop such a system.

5.1 Future Work

Assistive system using object recognition is in its very early stage and hence need lot of further efforts to make it feasible for end user, it may need industrial collaboration and funding. As far as future work is concern two main improvements would be useful, one is processing time reduction for object recognition module and second is more number of categories in dataset, in this contribution dataset contains 14 categories, so more number of categories can be added. The object recognition algorithm proposed here,

is highly parallel and hence the processing time can be improved using the techniques as below:

- DSP implementation with co-processors(eg.ARM), Davinci DSP board is appropriate choice.
- Using FPGA implemetaion, Vertex series.
- GPU implementation.

In future work few number of blind users can be hired, the system can be improved with testing on visually impaired people and there suggestions can be very helpful.

Publications

[1]. Rahul Kumar and Sukadev Meher."A Novel Method for Visually Impaired using Object Recognition." *Communications and Signal Processing (ICCSP), 2015 International Conference on, pages 775-779*, IEEE, 2015.

[2]. Rahul Kumar and Sukadev Meher. "Object Recognition based Assistive System for Visually Impaired". Robotics and Autonomous Systems, Elsevier Journal, ROBOT-D-15-00254(under review).

Bibliography

[1] Xiaodong Yang and Yingli Tian. Robust door detection in unfamiliar environments by combining edge and corner features. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 57–64. IEEE, 2010.

[2] Shuihua Wang and Yingli Tian. *Camera-Based signage detection and recognition for blind persons*. Springer, 2012.

[3] Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomaso Poggio. Robust object recognition with cortex-like mechanisms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(3):411–426, 2007.

[4] Nicolas Pinto, David D Cox, and James J DiCarlo. Why is real-world visual object recognition hard? *PLoS computational biology*, 4(1):e27, 2008.

[5] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[6] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[8] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] Adam Coates, Andrew Y Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011.

[11] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] Richard Socher, Brody Huval, Bharath Bath, Christopher D Manning, and Andrew Y Ng. Convolutional-recursive deep learning for 3d object classification. In *Advances in Neural Information Processing Systems*, pages 665–673, 2012.

[13] Judson P Jones and Larry A Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of neurophysiology*, 58(6):1233–1258, 1987.

[14] Mohammad Haghighat, Saman Zonouz, and Mohamed Abdel-Mottaleb. Identification using encrypted biometrics. In *Computer Analysis of Images and Patterns*, pages 440–448. Springer, 2013.

[15] World health organization: 10th revision of the who international statistical classification of diseases, injuries and causes of death.http://www.cdc.gov/nchs/icd/icd10.htm.

[16] Rabia Jafri, Syed Abid Ali, Hamid R Arabnia, and Shameem Fatima. Computer vision-based object recognition for the visually impaired in an indoors environment: a survey. *The Visual Computer*, 30(11):1197–1222, 2014.

[17] Suranga Nanayakkara, Roy Shilkrot, and Pattie Maes. Eyering: an eye on a finger. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, pages 1047–1050. ACM, 2012.

[18] Xiaodong Yang, Shuai Yuan, and YingLi Tian. Assistive clothing pattern recognition for visually impaired people. *Human-Machine Systems, IEEE Transactions on*, 44(2):234–243, 2014.

[19] Faiz M Hasanuzzaman, Xiaodong Yang, and YingLi Tian. Robust and effective component-based banknote recognition for the blind. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(6):1021–1030, 2012.

[20] Hiroyuki Takizawa, Satarou Yamaguchi, Masahiro Aoyagi, Nobuo Ezaki, and Seiya Mizuno. Kinect cane: an assistive system for the visually impaired based on three-dimensional object recognition. In *System Integration (SII), 2012 IEEE/SICE International Symposium on*, pages 740–745. IEEE, 2012.

[21] Vivek Pradeep, Gerard Medioni, and James Weiland. Robot vision for the visually impaired. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 15–22. IEEE, 2010.

[22] Pablo Vera, Daniel Zenteno, and Joaquín Salas. A smartphone-based virtual white cane. *Pattern Analysis and Applications*, 17(3):623–632, 2014.

[23] Wolfgang Fink and Mark Humayun. Digital object recognition audio-assistant for the visually impaired, January 5 2005. US Patent App. 11/030,678.

[24] Jeremi Sudol, Orang Dialameh, Chuck Blanchard, and Tim Dorcey. Looktel—a comprehensive platform for computer-aided visual assistance. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 73–80. IEEE, 2010.

[25] Tess Winlock, Eric Christiansen, and Serge Belongie. Toward real-time grocery detection for the visually impaired. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 49–56. IEEE, 2010.

[26] Diego López-de Ipiña, Tania Lorido, and Unai López. Blindshopping: enabling accessible shopping for visually impaired people through mobile technologies. In *Toward Useful Services for Elderly and People with Disabilities*, pages 266–270. Springer, 2011.

[27] Boris Schauerte, Manel Martinez, Angela Constantinescu, and Rainer Stiefelhagen. *An assistive vision system for the blind that helps find lost things.* Springer, 2012.

[28] Shuihua Wang and Yingli Tian. Detecting stairs and pedestrian crosswalks for the blind by rgbd camera. In *Bioinformatics and Biomedicine Workshops (BIBMW), 2012 IEEE International Conference on*, pages 732–739. IEEE, 2012.

[29] Hangrong Pan, Chucai Yi, and Yingli Tian. A primary travelling assistant system of bus detection and recognition for visually impaired people. In *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[30] Hugo Fernandes, Paulo Costa, Hugo Paredes, Vítor Filipe, and João Barroso. Integrating computer vision object recognition with location based services for the blind. In

Universal Access in Human-Computer Interaction. Aging and Assistive Environments, pages 493–500. Springer, 2014.

[31] Donatella Pascolini and Silvio Paolo Mariotti. Global estimates of visual impairment: 2010. *British Journal of Ophthalmology*, pages bjophthalmol–2011, 2011.

[32] Koen EA Van De Sande, Theo Gevers, and Cees GM Snoek. Evaluating color descriptors for object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1582–1596, 2010.

[33] Anthony J Bell and Terrence J Sejnowski. The "independent components" of natural scenes are edge filters. *Vision research*, 37(23):3327–3338, 1997.

[34]

[35] Simon S Haykin, Simon S Haykin, Simon S Haykin, and Simon S Haykin. *Neural networks and learning machines*, volume 3. Pearson Education Upper Saddle River, 2009.

[36] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. 2009.

[37] Kevin Jarrett, Koray Kavukcuoglu, M Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2146–2153. IEEE, 2009.

[38] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[39] Eben Upton and Gareth Halfacree. *Raspberry Pi user guide*. John Wiley & Sons, 2014.

[40] Gary Bradski. Learning-based computer vision with intel s open source computer vision library. *Intel Technology Journal*, (9), 2005.

[41] Peter BL Meijer. An experimental system for auditory image representations. *Biomedical Engineering, IEEE Transactions on*, 39(2):112–121, 1992.